



Cisco IOS IPv6 Configuration Guide

Release 12.4T

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Cisco IOS IPv6 Configuration Guide

© 2003 — 2009 Cisco Systems, Inc. All rights reserved.



About Cisco IOS and Cisco IOS XE Software Documentation

Last Updated: March 5, 2009

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

- [Documentation Objectives, page i](#)
- [Audience, page i](#)
- [Documentation Conventions, page ii](#)
- [Documentation Organization, page iii](#)
- [Additional Resources and Documentation Feedback, page xi](#)

Documentation Objectives

Cisco IOS documentation describes the tasks and commands available to configure and maintain Cisco networking devices.

Audience

The Cisco IOS documentation set is intended for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the configuration and maintenance tasks, the relationship among tasks, or the Cisco IOS commands necessary to perform particular tasks. The Cisco IOS documentation set is also intended for those users experienced with Cisco IOS who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS release.

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

- [Typographic Conventions, page ii](#)
- [Command Syntax Conventions, page ii](#)
- [Software Conventions, page iii](#)
- [Reader Alert Conventions, page iii](#)

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination ^D or Ctrl-D means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
<i>string</i>	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to <i>public</i> , do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
<i>italic</i>	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
...	An ellipsis (three consecutive nonbolded periods without spaces) after a syntax element indicates that the element can be repeated.
	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
Courier font	Courier font is used for information that is displayed on a PC or terminal screen.
Bold Courier font	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



Timesaver

Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

- [Cisco IOS Documentation Set, page iv](#)
- [Cisco IOS Documentation on Cisco.com, page iv](#)
- [Configuration Guides, Command References, and Supplementary Resources, page v](#)

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
 - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
 - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Cisco IOS XE, and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in [Table 2 on page xi](#).

Configuration Guides, Command References, and Supplementary Resources

[Table 1](#) lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at <http://www.cisco.com/web/psa/products/index.html>.

[Table 2](#) lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS AppleTalk Configuration Guide</i>	AppleTalk protocol.
<i>Cisco IOS XE AppleTalk Configuration Guide</i>	
<i>Cisco IOS AppleTalk Command Reference</i>	
<i>Cisco IOS Asynchronous Transfer Mode Configuration Guide</i>	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.
<i>Cisco IOS Asynchronous Transfer Mode Command Reference</i>	

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS Bridging and IBM Networking Configuration Guide</i> <i>Cisco IOS Bridging Command Reference</i> <i>Cisco IOS IBM Networking Command Reference</i>	<ul style="list-style-type: none"> Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
<i>Cisco IOS Broadband and DSL Configuration Guide</i> <i>Cisco IOS XE Broadband and DSL Configuration Guide</i> <i>Cisco IOS Broadband and DSL Command Reference</i>	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
<i>Cisco IOS Carrier Ethernet Configuration Guide</i> <i>Cisco IOS Carrier Ethernet Command Reference</i>	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
<i>Cisco IOS Configuration Fundamentals Configuration Guide</i> <i>Cisco IOS XE Configuration Fundamentals Configuration Guide</i> <i>Cisco IOS Configuration Fundamentals Command Reference</i>	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
<i>Cisco IOS DECnet Configuration Guide</i> <i>Cisco IOS XE DECnet Configuration Guide</i> <i>Cisco IOS DECnet Command Reference</i>	DECnet protocol.
<i>Cisco IOS Dial Technologies Configuration Guide</i> <i>Cisco IOS XE Dial Technologies Configuration Guide</i> <i>Cisco IOS Dial Technologies Command Reference</i>	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
<i>Cisco IOS Flexible NetFlow Configuration Guide</i> <i>Cisco IOS Flexible NetFlow Command Reference</i>	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS H.323 Configuration Guide</i>	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
<i>Cisco IOS High Availability Configuration Guide</i> <i>Cisco IOS XE High Availability Configuration Guide</i> <i>Cisco IOS High Availability Command Reference</i>	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
<i>Cisco IOS Integrated Session Border Controller Command Reference</i>	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
<i>Cisco IOS Intelligent Services Gateway Configuration Guide</i> <i>Cisco IOS Intelligent Services Gateway Command Reference</i>	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
<i>Cisco IOS Interface and Hardware Component Configuration Guide</i> <i>Cisco IOS XE Interface and Hardware Component Configuration Guide</i> <i>Cisco IOS Interface and Hardware Component Command Reference</i>	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
<i>Cisco IOS IP Addressing Services Configuration Guide</i> <i>Cisco IOS XE Addressing Services Configuration Guide</i> <i>Cisco IOS IP Addressing Services Command Reference</i>	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
<i>Cisco IOS IP Application Services Configuration Guide</i> <i>Cisco IOS XE IP Application Services Configuration Guide</i> <i>Cisco IOS IP Application Services Command Reference</i>	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
<i>Cisco IOS IP Mobility Configuration Guide</i> <i>Cisco IOS IP Mobility Command Reference</i>	Mobile ad hoc networks (MANet) and Cisco mobile networks.
<i>Cisco IOS IP Multicast Configuration Guide</i> <i>Cisco IOS XE IP Multicast Configuration Guide</i> <i>Cisco IOS IP Multicast Command Reference</i>	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS IP Routing Protocols Configuration Guide</i> <i>Cisco IOS XE IP Routing Protocols Configuration Guide</i> <i>Cisco IOS IP Routing Protocols Command Reference</i>	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
<i>Cisco IOS IP SLAs Configuration Guide</i> <i>Cisco IOS XE IP SLAs Configuration Guide</i> <i>Cisco IOS IP SLAs Command Reference</i>	Cisco IOS IP Service Level Agreements (IP SLAs).
<i>Cisco IOS IP Switching Configuration Guide</i> <i>Cisco IOS XE IP Switching Configuration Guide</i> <i>Cisco IOS IP Switching Command Reference</i>	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
<i>Cisco IOS IPv6 Configuration Guide</i> <i>Cisco IOS XE IPv6 Configuration Guide</i> <i>Cisco IOS IPv6 Command Reference</i>	For IPv6 features, protocols, and technologies, go to the IPv6 “Start Here” document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
<i>Cisco IOS ISO CLNS Configuration Guide</i> <i>Cisco IOS XE ISO CLNS Configuration Guide</i> <i>Cisco IOS ISO CLNS Command Reference</i>	ISO connectionless network service (CLNS).
<i>Cisco IOS LAN Switching Configuration Guide</i> <i>Cisco IOS XE LAN Switching Configuration Guide</i> <i>Cisco IOS LAN Switching Command Reference</i>	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
<i>Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide</i> <i>Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference</i>	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
<i>Cisco IOS Mobile Wireless Home Agent Configuration Guide</i> <i>Cisco IOS Mobile Wireless Home Agent Command Reference</i>	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
<i>Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide</i> <i>Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference</i>	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
<i>Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide</i> <i>Cisco IOS Mobile Wireless Radio Access Networking Command Reference</i>	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS Multiprotocol Label Switching Configuration Guide</i> <i>Cisco IOS XE Multiprotocol Label Switching Configuration Guide</i> <i>Cisco IOS Multiprotocol Label Switching Command Reference</i>	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
<i>Cisco IOS Multi-Topology Routing Configuration Guide</i> <i>Cisco IOS Multi-Topology Routing Command Reference</i>	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
<i>Cisco IOS NetFlow Configuration Guide</i> <i>Cisco IOS XE NetFlow Configuration Guide</i> <i>Cisco IOS NetFlow Command Reference</i>	Network traffic data analysis, aggregation caches, export features.
<i>Cisco IOS Network Management Configuration Guide</i> <i>Cisco IOS XE Network Management Configuration Guide</i> <i>Cisco IOS Network Management Command Reference</i>	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
<i>Cisco IOS Novell IPX Configuration Guide</i> <i>Cisco IOS XE Novell IPX Configuration Guide</i> <i>Cisco IOS Novell IPX Command Reference</i>	Novell Internetwork Packet Exchange (IPX) protocol.
<i>Cisco IOS Optimized Edge Routing Configuration Guide</i> <i>Cisco IOS Optimized Edge Routing Command Reference</i>	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
<i>Cisco IOS Quality of Service Solutions Configuration Guide</i> <i>Cisco IOS XE Quality of Service Solutions Configuration Guide</i> <i>Cisco IOS Quality of Service Solutions Command Reference</i>	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
<i>Cisco IOS Security Configuration Guide</i> <i>Cisco IOS XE Security Configuration Guide</i> <i>Cisco IOS Security Command Reference</i>	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
<i>Cisco IOS Service Selection Gateway Configuration Guide</i> <i>Cisco IOS Service Selection Gateway Command Reference</i>	Subscriber authentication, service access, and accounting.
<i>Cisco IOS Software Activation Configuration Guide</i> <i>Cisco IOS Software Activation Command Reference</i>	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
<i>Cisco IOS Software Modularity Installation and Configuration Guide</i> <i>Cisco IOS Software Modularity Command Reference</i>	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
<i>Cisco IOS Terminal Services Configuration Guide</i> <i>Cisco IOS Terminal Services Command Reference</i> <i>Cisco IOS XE Terminal Services Command Reference</i>	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
<i>Cisco IOS Virtual Switch Command Reference</i>	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). Note For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
<i>Cisco IOS Voice Configuration Library</i> <i>Cisco IOS Voice Command Reference</i>	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
<i>Cisco IOS VPDN Configuration Guide</i> <i>Cisco IOS XE VPDN Configuration Guide</i> <i>Cisco IOS VPDN Command Reference</i>	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
<i>Cisco IOS Wide-Area Networking Configuration Guide</i> <i>Cisco IOS XE Wide-Area Networking Configuration Guide</i> <i>Cisco IOS Wide-Area Networking Command Reference</i>	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
<i>Cisco IOS Wireless LAN Configuration Guide</i> <i>Cisco IOS Wireless LAN Command Reference</i>	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
<i>Cisco IOS Master Command List, All Releases</i>	Alphabetical list of all the commands documented in all Cisco IOS releases.
<i>Cisco IOS New, Modified, Removed, and Replaced Commands</i>	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
<i>Cisco IOS Software System Messages</i>	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
<i>Cisco IOS Debug Command Reference</i>	Alphabetical list of debug commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2008–2009 Cisco Systems, Inc. All rights reserved.



Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last Updated: March 5, 2009

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

- [Initially Configuring a Device, page i](#)
- [Using the CLI, page ii](#)
- [Saving Changes to a Configuration, page xii](#)
- [Additional Information, page xii](#)

For more information about using the CLI, see the “[Using the Cisco IOS Command-Line Interface](#)” section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the “[About Cisco IOS and Cisco IOS XE Software Documentation](#)” document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at <http://www.cisco.com/web/psa/products/index.html>.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

**Note**

The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

- [Understanding Command Modes, page ii](#)
- [Using the Interactive Help Feature, page v](#)
- [Understanding Command Syntax, page vi](#)
- [Understanding Enable and Enable Secret Passwords, page viii](#)
- [Using the Command History Feature, page viii](#)
- [Abbreviating Commands, page ix](#)
- [Using Aliases for CLI Commands, page ix](#)
- [Using the no and default Forms of Commands, page x](#)
- [Using the debug Command, page x](#)
- [Filtering Output Using Output Modifiers, page x](#)
- [Understanding CLI Error Messages, page xi](#)

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

[Table 1](#) lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 *CLI Command Modes*

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	Router>	Issue the logout or exit command.	<ul style="list-style-type: none"> • Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the enable command.	Router#	Issue the disable command or the exit command to return to user EXEC mode.	<ul style="list-style-type: none"> • Issue show and debug commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the configure terminal command.	Router(config)#	Issue the exit command or the end command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the interface command.	Router(config-if)#	Issue the exit command to return to global configuration mode or the end command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the line vty or line console command.	Router(config-line)#	Issue the exit command to return to global configuration mode or the end command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the reload command. Press the Break key during the first 60 seconds while the system is booting.	rommon # > The # symbol represents the line number and increments at each prompt.	Issue the continue command.	<ul style="list-style-type: none"> Run as the default operating mode when a valid image cannot be loaded. Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	<p>The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload.</p> <ul style="list-style-type: none"> A user-configured access policy was configured using the transport-map command, which directed the user into diagnostic mode. The router was accessed using an RP auxiliary port. A break signal (Ctrl-C, Ctrl-Shift-6, or the send break command) was entered, and the router was configured to enter diagnostic mode when the break signal was received. 	Router(diag)#	<p>If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode.</p> <p>If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI.</p> <p>If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.</p>	<ul style="list-style-type: none"> Inspect various states on the router, including the Cisco IOS state. Replace or roll back the configuration. Provide methods of restarting the Cisco IOS software or other processes. Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias                set and display aliases command
boot                 boot up an external process
confreg              configuration register utility
cont                  continue executing a downloaded image
context              display the context of a loaded image
cookie               display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```



Note

A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. [Table 2](#) describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
<i>partial command?</i>	Provides a list of commands that begin with the character string (no space between the command and the question mark).
<i>partial command</i> <Tab>	Completes a partial command name (no space between the command and <Tab>).
<i>command ?</i>	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
<i>command keyword ?</i>	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help
```

Help may be requested at any point in a command by entering a question mark '?'. If nothing matches, the help list will be empty and you must backup until entering a '?' shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?') and describes each possible argument.
2. Partial help is provided when an abbreviated argument is entered and you want to know what arguments match the input (e.g. 'show pr?').

?

```
Router# ?
```

Exec commands:

access-enable	Create a temporary access-List entry
access-profile	Apply user-profile to interface
access-template	Create a temporary access-List entry
alps	ALPS exec commands
archive	manage archive files

```
<snip>
```

partial command?

```
Router(config)# zo?
```

```
zone zone-pair
```

partial command<Tab>

```
Router(config)# we<Tab> webvpn
```

command?

```
Router(config-if)# pppoe ?
```

enable	Enable pppoe
max-sessions	Maximum PPPOE sessions

command keyword?

```
Router(config-if)# pppoe enable ?
```

group	attach a BBA group
-------	--------------------

```
<cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. [Table 3](#) describes these conventions.

Table 3 *CLI Syntax Conventions*

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```

Router(config)# ethernet cfm domain ?
WORD domain name
Router(config)# ethernet cfm domain dname ?
level
Router(config)# ethernet cfm domain dname level ?
<0-7> maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
<cr>
Router(config)# snmp-server file-transfer access-group 10 ?
protocol protocol options
<cr>
Router(config)# logging host ?
Hostname or A.B.C.D IP address of the syslog server
ipv6 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
protocol protocol options
<cr>

```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, “two words” is a valid password. Leading spaces are ignored, but trailing spaces are recognized.



Note

Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.



Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or un	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias mode command-alias original-command**. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see

http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebg all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at

http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.



Caution

Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin regular expression**—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include regular expression**—Displays all lines in which a match of the regular expression is found.
- **exclude regular expression**—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression “protocol.”

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. [Table 5](#) shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: “show con”	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at “^” marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- [Cisco IOS Release 12.2SR System Message Guide](#)
- [Cisco IOS System Messages, Volume 1 of 2](#) (Cisco IOS Release 12.4)
- [Cisco IOS System Messages, Volume 2 of 2](#) (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- “Using the Cisco IOS Command-Line Interface” section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:
http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html
or
“Using Cisco IOS XE Software” chapter of the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*:
http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/Using_CLI.html
- Cisco Product Support Resources
<http://www.cisco.com/web/psa/products/index.html>
- Support area on Cisco.com (also search for documentation by task or product)
<http://www.cisco.com/en/US/support/index.html>
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)
<http://www.cisco.com/kobayashi/sw-center/>
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software
<http://www.cisco.com/cgi-bin/Support/Errordecoder/index.cgi>
- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)
<http://tools.cisco.com/Support/CLILookup>

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

<https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl>

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

© 2008–2009 Cisco Systems, Inc. All rights reserved.



Start Here: Cisco IOS Software Release Specifics for IPv6 Features

This document lists the IP version 6 (IPv6) features supported in the 12.0S, 12.xT, 12.2S family, XE, 12.3, and 12.4 Cisco IOS software release trains.

The IPv6 for Cisco IOS Software feature documentation provides implementation and command reference information for IPv6 features supported in the Cisco IOS software. This Start Here document details only the Cisco IOS software release specifics for IPv6 features. Not all IPv6 features may be supported in your Cisco IOS software release. We strongly recommend that you read this entire document before reading the other IPv6 for Cisco IOS software feature documentation.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

The *Cisco IOS IPv6 Configuration Guide* is located at the following website:

<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide>

The *Cisco IOS IPv6 Command Reference* is located at the following website:

<http://www.cisco.com/en/US/docs/ios/ipv6/command/reference>

The following sections are included in this document:

- [Cisco IOS Software Platform Dependencies and Restrictions, page 1](#)
- [Cisco IOS IPv6 Features and Supported Software Releases, page 2](#)
- [Cisco Platforms Supporting IPv6 Hardware Forwarding, page 19](#)
- [Additional References, page 21](#)

Cisco IOS Software Platform Dependencies and Restrictions

IPv6 features are supported in the 12.0S, 12.xT, 12.2S, 12.2SB, 12.2SE, 12.2SG, 12.2SR, 12.2SX, XE, 12.3, and 12.4 Cisco IOS software release trains, starting at Cisco IOS Release 12.0(22)S, 12.2(2)T, 12.2(14)S, 12.2(28)SB, 12.2(25)SEA, 12.2(33)SRA, 12.2(17a)SX1, Cisco IOS XE Release 2.1, 12.3,



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006—2009 Cisco Systems, Inc. All rights reserved.

and 12.4, respectively. See [Table 1](#) to determine which IPv6 features are supported in each release of the 12.0S, 12.xT, 12.2S, 12.2SB, 12.2SR, 12.2SX, Cisco IOS XE Release 2.1, 12.3, and 12.4 Cisco IOS software trains.

- IPv6 was introduced on the 12.0(21)ST Cisco IOS software release train, which was merged with the 12.0S Cisco IOS software release train starting at Cisco IOS Release 12.0(22)S. The 12.0S Cisco IOS software release train provides IPv6 support on Cisco 12000 series Internet routers and Cisco 10720 Internet routers only.
- The 12.2S Cisco IOS release train comprises a family of release trains, each supporting different platforms as follows:
 - The 12.2SB Cisco IOS release train comprises the Cisco 10000, 7304, 7301, and 7200 series. As of Cisco IOS Release 12.2(33)SB, the Cisco 7200 and 7301 series are not supported on the 12.2SB release train.
 - The 12.2SE Cisco IOS release train consists of the Cisco Catalyst 3560, 3750, 3560E, and 3750E series.
 - The 12.2SG Cisco IOS release train consists of the Cisco Catalyst 4500 and Cisco Catalyst 4900 series.
 - The 12.2SR Cisco IOS release train consists of the Cisco 7600 and 7200 series routers. In addition, Cisco IOS XE on ASR 1000 series routers inherits IPv6 features from 12.2SR releases but does not support all features on this train.
 - The 12.2SX Cisco IOS release train consists of the Cisco Catalyst 6500. Before the 12.2SR Cisco IOS release train, the 12.2SX release train also included the Cisco 7600 series.
- IPv6 is also supported in some special software release trains.

Cisco IOS IPv6 Features and Supported Software Releases

[Table 1](#) lists the IPv6 features supported in the 12.0S, 12.xT, 12.2S, 12.2SB, 12.2SR, 12.2SX, Cisco IOS XE Release 2.1, 12.3, and 12.4 Cisco IOS software release trains.



Note

[Table 1](#) identifies the earliest release for each software release train in which the feature became available. Unless noted otherwise in [Table 1](#), subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Supported IPv6 Feature

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6		(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 address types: Unicast	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	—	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6: uRPF	Implementing IPv6 Addressing and Basic Connectivity	(31)	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
IPv6: ICMPv6	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28) ¹	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: IPv6 neighbor discovery	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: IPv6 stateless autocon-figuration	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: IPv6 MTU path discovery	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: ping	Implementing IPv6 for Network Management	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6:	Implementing IPv6 Addressing and Basic Connectivity, Implementing IPv6 for Network Management	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: ICMPv6 redirect	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(4)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6: ICMP rate limiting	Implementing IPv6 Addressing and Basic Connectivity	—	12.2(8)	12.3	—	—	—	—	—	Cisco IOS XE Release 2.1
IPv6: neighbor discovery duplicate address detection	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(4)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6: IPv6 static cache entry for neighbor discovery	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(8)	12.3	(28)	(25)SEA	—	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 address types: Anycast	Implementing IPv6 Addressing and Basic Connectivity	—	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6: NetFlow for IPv6 unicast traffic	Implementing NetFlow for IPv6	—	12.3(7)	12.4	—	—	—	(33)SRB	(33)SXH	—
IPv6: NetFlow: Flexible NetFlow for IPv6 replaces IPv6 NetFlow	Implementing NetFlow for IPv6	—	12.4(20)	—	—	—	—	—	—	—
IPv6: Mobile IPv6 home agent	Implementing Mobile IPv6	—	12.3(14)	12.4	—	—	—	—	—	—
IPv6: IPv6 default router preferences	Implementing IPv6 Addressing and Basic Connectivity	—	12.4(2)	—	(33)	—	—	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6: IPv6 ACL extensions for Mobile IPv6	Implementing Mobile IPv6	—	12.4(2)	—	—	—	—	(33)SRB	(33)SXI	—
IPv6: Mobile IP - Mobile v6 - Basic NEMO	Implementing Mobile IPv6	—	12.4(20)	—	—	—	—	—	—	—
IPv6: IP Receive ACL for IPv6 traffic	IP Receive ACL	(32)	—	—	—	—	—	—	—	—

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6: syslog over IPv6	Implementing IPv6 for Network Management	—	12.4(4)	—	(33)	—	—	(33)SRB	(33)SXI	Cisco IOS XE Release 2.1
IPv6: IPv6 VPN over MPLS	Implementing IPv6 VPN over MPLS	—	12.4(20)	—	(33)	—	—	(33)SRB	(33)SXI	—
IPv6: MPLS VPN 6VPE support over IP tunnels	Implementing IPv6 VPN over MPLS (6VPE)	—	—	—	—	—	—	(33)SRB1	(33)SXI	—
IPv6: CNS agents for IPv6	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	—
IPv6: IP SLAs for IPv6	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6: IPv6 for config logger	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6: IPv6 Netconf support	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	—
IPv6: IPv6 support for TCL	Implementing IPv6 for Network Management	—	12.4(20)	—	—	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6: IPv6 support in SOAP	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6: HTTP(S) IPv6 support (Infrastructure)	Implementing IPv6 for Network Management	—	12.4(20)	—	(33)	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6: no ipv6 source-route command	Cisco IOS IPv6 Command Reference	—	12.3(4)	12.4	—	—	—	(33)SRB1	—	—

IPv6 Switching Services

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 switching: automatic 6to4 tunnels	Implementing Tunneling for IPv6	(22) ²	12.2(2)	12.3	(28)SB	—	—	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 switching: Cisco Express Forwarding/distributed Cisco Express Forwarding support	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(13)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 switching: CEFv6 switched configured IPv6 over IPv4 tunnels	Implementing Tunneling for IPv6	—	12.2(13)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	—
IPv6 switching: provider edge router over MPLS (6PE) ^{3 4}	Implementing IPv6 over MPLS	(22)	12.2(15)	12.3	(31)	—	—	(33)SRA	(17b)SXA	—
IPv6 switching: CEFv6 switched ISATAP tunnels	Implementing Tunneling for IPv6	—	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 switching: CEFv6 switched automatic IPv4-compatible tunnels	Implementing Tunneling for IPv6	—	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1

IPv6 Routing

IPv6 routing: RIP for IPv6 (RIPng)	Implementing RIP for IPv6	(22)	12.2(2) ⁵	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 routing: static routing	Implementing Static Routes for IPv6	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 routing: route redistribution	Implementing IS-IS for IPv6 , Implementing RIP for IPv6	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 routing: multiprotocol BGP extensions for IPv6	Implementing Multiprotocol BGP for IPv6	(22)	12.2(2) ⁶	12.3	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 routing: multiprotocol BGP link-local address peering	Implementing Multiprotocol BGP for IPv6	(22)	12.2(4)	12.3	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 routing: IS-IS support for IPv6	Implementing IS-IS for IPv6	(22)	12.2(8)	12.3	(28)	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 routing: IS-IS multitopology support for IPv6	Implementing IS-IS for IPv6	(26)	12.2(15)	12.3	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 routing: OSPF for IPv6 (OSPFv3)	Implementing OSPF for IPv6	(24)	12.2(15)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 routing: OSPF for IPv6 authentication support with IPsec	Implementing OSPF for IPv6	—	12.3(4)	12.4	—	—	—	—	—	—
IPv6 Routing: OSPF IPv6 (OSPFv3) IPsec ESP Encryption and Authentication	Implementing OSPF for IPv6	—	12.4(9)	—	—	—	—	—	—	—
OSPFv3 dynamic interface cost support	Implementing OSPF for IPv6	—	12.4(15)	—	—	—	—	—	—	—
IPv6 routing: IPv6 policy-based routing	Implementing Policy-Based Routing for IPv6	—	12.3(7)	12.4	—	—	—	—	—	—

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 routing: EIGRP support	Implementing EIGRP for IPv6	—	12.4(6)	—	—	(40)SE	—	(33)SRB	(33)SXI	Cisco IOS XE Release 2.2
IPv6 routing: OSPFv3 Fast Convergence - LSA and SPF throttling	Implementing OSPF for IPv6	—	—	—	(33)	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
OSPFv3 graceful restart	Implementing OSPF for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
OSPFv3 for BFD	Implementing OSPF for IPv6, Implementing Bidirectional Forwarding Detection for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
Static Route support for BFD over IPv6	Implementing Bidirectional Forwarding Detection for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1

IPv6 Services and Management

IPv6 services: AAAA DNS lookups over an IPv4 transport	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(17a)SXI	Cisco IOS XE Release 2.1
IPv6 services: standard access control lists	Implementing Traffic Filters and Firewalls for IPv6 Security	(22)	12.2(2)	12.3	(28)	(25)SED	(25)	(33)SRA	(17a)SXI	Cisco IOS XE Release 2.1
IPv6 services: IPv6 ACL extensions for IPsec authentication header	Implementing Traffic Filters and Firewalls for IPv6 Security	—	12.4(20)	—	—	—	—	—	—	—
IPv6 services: DNS lookups over an IPv6 transport	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(8)	12.3	(28)	(25)SED	(25)	(33)SRA	(17a)SXI	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 services: Secure Shell (SSH) support over IPv6	Implementing IPv6 for Network Management	(22)	12.2(8)	12.3	(28)	(25)SEE	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 services: Cisco Discovery Protocol—IPv6 address family support for neighbor information	Implementing IPv6 Addressing and Basic Connectivity	—	12.2(8)	12.3	(28)	(25)SEE	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 services: CISCO-IP-MIB support	Implementing IPv6 for Network Management	(22)	12.2(15)	12.3	(28)	(25)SEE	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 services: CISCO-IP-FO RWARDING-MIB support	Implementing IPv6 for Network Management	(22)	12.2(15)	12.3	(28)	(25)SEE	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 services: IP tunnel MIB support	Implementing IPv6 Addressing and Basic Connectivity	—	—	—	—	—	—	(33)SRB	—	—
IPv6 services: RFC 4293 IP-MIB (IPv6 only) and RFC 4292 IP-FORWARD-MIB (IPv6 only)	Implementing IPv6 for Network Management	—	—	—	—	—	—	(33)SRC	—	Cisco IOS XE Release 2.1
IPv6 services: extended access control lists ⁴	Implementing Traffic Filters and Firewalls for IPv6 Security	(23)	12.2(13)	12.3	(28)	(25)SED	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 services: generic prefix	Implementing IPv6 Addressing and Basic Connectivity	—	12.3(4)	12.4	—	—	—	—	—	—
IPv6 services: SNMP over IPv6 ⁷	Implementing IPv6 for Network Management	(27)	12.3(14)	12.4	(33)	—	—	(33)SRB	(33)SXI	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
SNMPv3 - 3DES and AES Encryption Support	Implementing IPv6 for Network Management	—	12.4(2)	—	(33)	—	—	(33)SRB	(33)SXI	—
IPv6 services: IPv6 IOS Firewall	Implementing Traffic Filters and Firewalls for IPv6 Security	—	12.3(7)	12.4	—	—	—	—	—	—
IPv6 services: IPv6 IOS Firewall FTP application support	Implementing Traffic Filters and Firewalls for IPv6 Security	—	12.3(11)	—	—	—	—	—	—	—
IPv6 services: IPv6 IPsec VPN	Implementing IPsec in IPv6 Security	—	12.4(4)	—	—	—	—	—	—	—
IPv6 Secure Neighbor Discovery (SeND)	Implementing Secure Neighbor Discovery in IPv6	—	12.4(24)	—	—	—	—	—	—	—
IPv6 services: IPv6 over DMVPN	Implementing Dynamic Multipoint VPN over IPv6	—	12.4(20)	—	—	—	—	—	—	—
IPv6 services: HSRP for IPv6	Configuring First Hop Redundancy Protocols in IPv6	—	12.4(4)	—	—	—	—	(33)SRB	(33)SXI	Cisco IOS XE Release 2.2
IPv6 services: FHRP - GLBP for IPv6	Configuring First Hop Redundancy Protocols in IPv6	—	12.4(6)	—	—	—	—	—	(33)SXI	—
IPv6 over Frame Relay	Implementing IPv6 over Frame Relay	(33)	—	—	—	—	—	—	—	—

IPv6 Broadband Access

IPv6 access services: PPPoA	Implementing ADSL and Deploying Dial Access for IPv6	—	12.2(13)	12.3	—	—	—	—	—	—
-----------------------------	--	---	----------	------	---	---	---	---	---	---

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 access services: PPPoE	Implementing ADSL and Deploying Dial Access for IPv6	—	12.2(13)	12.3	—	—	—	—	—	—
IPv6 access services: prefix pools	Implementing ADSL and Deploying Dial Access for IPv6	—	12.2(13)	12.3	—	—	—	—	—	—
IPv6 access services: AAA support for Cisco VSA IPv6 attributes	Implementing ADSL and Deploying Dial Access for IPv6	—	12.2(13)	12.3	—	—	—	—	—	—
IPv6 access services: remote bridged encapsulation	Implementing IPv6 Addressing and Basic Connectivity	—	12.3(4)	12.4	—	—	—	—	—	—
IPv6 access services: AAA support for RFC 3162 IPv6 RADIUS attributes	Implementing ADSL and Deploying Dial Access for IPv6	—	12.3(4)	12.4	—	—	—	—	—	—
DHCP for IPv6										
IPv6 access services: stateless DHCPv6	Implementing DHCP for IPv6	(32) ⁸	12.3(4)	12.4	(28)	—	—	(33)SRA	(18)SXE	—
IPv6 access services: DHCPv6 prefix delegation	Implementing DHCP for IPv6, Implementing ADSL and Deploying Dial Access for IPv6	(32) ⁸	12.3(4)	12.4	(28)	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 access services: DHCP for IPv6 relay agent	Implementing DHCP for IPv6	—	12.3(11)	12.4	(28)	(46)	(50)	(33)SRC	(33)SXI	—

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 access services: DHCPv6 prefix delegation via AAA	Implementing ADSL and Deploying Dial Access for IPv6	—	12.3(14)	12.4	(28) ⁹	—	—	—	—	—
IPv6 access services: DHCPv6 Server Stateless Auto Configuration	Implementing DHCP for IPv6	—	12.4(15)	—	—	—	—	—	—	—
IPv6 access services: DHCPv6 Client Information Refresh Option	Implementing DHCP for IPv6	—	12.4(15)	—	—	—	—	—	—	—
IPv6 access services: DHCPv6 relay agent notification for prefix delegation ¹⁰	Implementing DHCP for IPv6	—	—	—	—	(46)	—	(33)SRC	(33)SXI	Cisco IOS XE Release 2.1
IPv6 access services: DHCPv6 relay - reload persistent interface ID option	Implementing DHCP for IPv6	—	—	—	(33)	(46)	—	(33)SRC	(33)SXI	Cisco IOS XE Release 2.1
IPv6 access services: DHCPv6 Ethernet remote ID option	Implementing DHCP for IPv6	—	—	—	—	(46)	—	(33)SRC	(33)SXI	Cisco IOS XE Release 2.1
DHCP - DHCPv6 Individual Address Assignment	Implementing DHCP for IPv6	—	12.4(24)	—	—	(46)	—	—	—	—

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 Multicast		(26) ¹¹	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: Multicast Listener Discovery (MLD) protocol, versions 1 and 2	Implementing IPv6 Multicast	(26) ¹¹	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: PIM sparse mode (PIM-SM)	Implementing IPv6 Multicast	(26) ¹¹	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: PIM Source Specific Multicast (PIM-SSM)	Implementing IPv6 Multicast	(26) ¹¹	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: scope boundaries	Implementing IPv6 Multicast	(26) ¹¹	12.3(2)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	—
IPv6 multicast: MLD access group	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: PIM accept register	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: PIM embedded RP support	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: RPF flooding of bootstrap router (BSR) packets	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: routable address hello option	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 multicast: static multicast routing (mroute)	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	—	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: address family support for multiprotocol Border Gateway Protocol (MBGP)	Implementing IPv6 Multicast	(26) ¹¹	12.3(4)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: Explicit tracking of receivers	Implementing IPv6 Multicast	—	12.3(7)	12.4	(28)	—	(25)	(33)SRA	(33)SXH	Cisco IOS XE Release 2.1
IPv6 multicast: IPv6 bidirectional PIM	Implementing IPv6 Multicast	—	12.3(7)	12.4	(28)	—	(25)	(33)SRA	—	Cisco IOS XE Release 2.1
IPv6 multicast: MFIB display enhancements	Implementing IPv6 Multicast	—	12.3(7)	12.4	—	—	—	—	—	—
IPv6 multicast: IPv6 BSR	Implementing IPv6 Multicast	(28)	12.3(11)	12.4	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: IPv6 BSR bidirectional support	Implementing IPv6 Multicast	—	12.3(14)	12.4	—	—	—	—	—	—
IPv6 multicast: IPv6 BSR scoped-zone support	Implementing IPv6 Multicast	—	—	—	—	—	—	—	(18)SXE	—
IPv6 multicast: SSM mapping for MLDv1 SSM	Implementing IPv6 Multicast	—	12.4(2)	—	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 multicast: IPv6 BSR—ability to configure RP mapping	Implementing IPv6 Multicast	—	12.4(2)	—	—	—	—	—	—	—

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 multicast: MLD group limits	Implementing IPv6 Multicast	—	12.4(2)	—	—	—	—	—	—	—
IPv6 multicast: multicast user authentication and profile support	Implementing IPv6 Multicast	—	12.4(4)	—	—	—	—	—	—	—
IPv6 multicast: MLD snooping	Implementing IPv6 Multicast	—	—	—	—	(25)SED	—	(33)SRA	(18)SXE	—
IPv6 multicast: Address Group Range Support	Implementing IPv6 Multicast	—	—	—	—	—	—	—	(33)SXI	—
NAT Protocol Translation (NAT-PT)		—	12.2(13)	12.3	—	—	—	—	—	—
NAT-PT: support for DNS ALG	Implementing NAT Protocol Translation	—	12.2(13)	12.3	—	—	—	—	—	—
NAT-PT: support for overload (PAT)	Implementing NAT Protocol Translation	—	12.3(2)	12.4	—	—	—	—	—	—
NAT-PT: support for FTP ALG	Implementing NAT Protocol Translation	—	12.3(2)	12.4	—	—	—	—	—	—
NAT-PT: support for fragmentation	Implementing NAT Protocol Translation	—	12.3(2)	12.4	—	—	—	—	—	—
IPv6 Tunnel Services										
IPv6 tunneling: automatic 6to4 tunnels	Implementing Tunneling for IPv6	(22)	12.2(2)	12.3	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 tunneling: automatic IPv4-compatible tunnels	Implementing Tunneling for IPv6	(22)	12.2(2)	12.3	(28)	—	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 tunneling: manually configured IPv6 over IPv4 tunnels	Implementing Tunneling for IPv6	(23) ²	12.2(2)	12.3	(28)	—	(25)	(33)SRA	(17a)SXI	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 tunneling: IPv6 over IPv4 GRE tunnels	Implementing Tunneling for IPv6	(22) ¹²	12.2(4)	12.3	(28)	—	—	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 tunneling: IPv6 over UTI using a tunnel line card ¹³	Implementing Tunneling for IPv6	(23) ²	—	—	—	—	—	—	—	—
IPv6 tunneling: ISATAP tunnel support	Implementing Tunneling for IPv6	—	12.2(15)	12.3	(28)SB	—	(25)	(33)SRA	(17a)SX1	Cisco IOS XE Release 2.1
IPv6 tunneling: IPv6 over IPv6 tunnels	Implementing Tunneling for IPv6	—	12.3(7)	12.4	—	—	—	—	—	—
IPv6 tunneling: IP over IPv6 GRE tunnels	Implementing Tunneling for IPv6	—	12.3(7)	12.4	—	—	—	—	—	—
IPv6 tunneling: IPv6 GRE tunnels in CLNS	Implementing Tunneling for IPv6	—	12.3(7)	12.4	(28)SB	—	—	(33)SRA	—	Cisco IOS XE Release 2.1
IPv6 QoS (Quality of Service)		(28)	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 QoS: MQC packet classification	Implementing QoS for IPv6	—	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 QoS: MQC traffic shaping	Implementing QoS for IPv6	(28)	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 QoS: MQC traffic policing	Implementing QoS for IPv6	(28)	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 QoS: MQC packet marking/re-marking	Implementing QoS for IPv6	(28)	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 QoS: queueing	Implementing QoS for IPv6	—	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 QoS: MQC weighted random early detection (WRED)-based drop	Implementing QoS for IPv6	(28)	12.2(13)	12.3	—	—	—	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1

IPv6 High Availability

IPv6: Base protocols High Availability	Implementing IPv6 Addressing and Basic Connectivity	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
IPv6 routing: RIPng nonstop forwarding	Implementing RIP for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
IPv6 routing: NSF and graceful restart for MP-BGP IPv6 address family	Implementing Multiprotocol BGP for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1
OSPFv3 graceful restart	Implementing OSPF for IPv6	—	—	—	—	—	—	—	—	Cisco IOS XE Release 2.1

IPv6 Voice

RTP/RTCP over IPv6	Implementing Voice over IPv6	—	12.4(22)	—	—	—	—	—	—	—
--------------------	--	---	----------	---	---	---	---	---	---	---

IPv6 Data Link Layer

IPv6 data link: ATM PVC and ATM LANE	Implementing IPv6 Addressing and Basic Connectivity	(22) ¹⁴	12.2(2)	12.3	(28)	—	—	(33)SRA	—	—
IPv6 data link: Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	—	(33)SRA	—	Cisco IOS XE Release 2.1

Feature	Where Documented	12.0S Release	12.xT Release	12.x Release	12.2SB Release	12.2SE Release	12.2SG Release	12.2SR Release	12.2SX Release	XE Release
IPv6 data link: Frame Relay PVC	Implementing IPv6 Addressing and Basic Connectivity	(22) ¹⁵	12.2(2)	12.3	(28)	—	—	(33)SRA	—	Cisco IOS XE Release 2.1
IPv6 data link: High-Level Data Link Control	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	—	—	(33)SRA	—	Cisco IOS XE Release 2.1
IPv6 data link: PPP service over packet over SONET, ISDN, and serial (synchronous and asynchronous) interfaces	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28) ¹⁶	—	—	(33)SRA	—	Cisco IOS XE Release 2.1
IPv6 data link: VLANs using IEEE 802.1Q encapsulation	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 data link: VLANs using Cisco Inter-Switch Link (ISL)	Implementing IPv6 Addressing and Basic Connectivity	(22)	12.2(2)	12.3	(28)	(25)SEA	(25)	(33)SRA	(18)SXE	Cisco IOS XE Release 2.1
IPv6 data link: dynamic packet transport (DPT)	Implementing IPv6 Addressing and Basic Connectivity	(23)	—	—	—	—	—	—	—	—

1. A ping in the fast-path mode is not supported. The support rate is limited to 10 pings per second per interface.
2. In Cisco IOS Release 12.0(23)S, the Cisco 12000 series Internet router provides enhanced performance for IPv6 manually configured tunnels by processing traffic on the line card.
3. The Cisco 10720 Internet router is supported in Cisco IOS Release 12.0(26)S.
4. IPv6 extended access control lists and IPv6 provider edge routers over MPLS are implemented with IPv6 hardware acceleration on the Cisco 12000 series Internet router IP service engine (ISE) line cards in Cisco IOS routers in Cisco IOS Release 12.0(25)S and later releases.
5. The RIP for IPv6 feature was updated in Cisco IOS Release 12.2(13)T.
6. Enhancements were made to several multiprotocol BGP commands.
7. SNMP versions 1, 2, and 3 are supported over an IPv6 transport.
8. In Cisco IOS Release 12.0(32)S, the Dynamic Host Configuration Protocol (DHCP) for IPv6 prefix delegation is supported on shared port adaptors (SPAs) in the 10G Engine 5 SPA Interface Processor (SIP) on the Cisco 12000 series Internet router only for stateless address assignment.
9. This feature may not be useful without either the IPv6 over PPPoE or IPv6 over PPPoA features, and the IPv6 over PPPoE or IPv6 over PPPoA features are not supported in Cisco IOS Release 12.2(28)SB.

10. Support for this feature is provided in Cisco IOS Release 12.2(33)SCA (see [Table 3](#)).
11. Feature is supported on Cisco 12000 series Internet routers in Cisco IOS Release 12.0(26)S.
12. IPv6 over IPv4 GRE tunnels are not supported on the Cisco 12000 series Internet router.
13. Feature is supported on the Cisco 12000 series Internet router only.
14. Only ATM PVCs are supported on the Cisco IOS 2.0S software release train. ATM LANE is not supported.
15. Frame Relay PVCs are not supported by distributed Cisco Express Forwarding switching for IPv6 in the 12.0S Cisco IOS software train. In the Cisco 12000 series Internet routers, Frame Relay encapsulated IPv6 packets are process switched on the Route Processor.
16. In 12.2(28)SB, PPPoA, PPPoE, and PPP over a VLAN are not supported. PPP over a serial link is supported.

Cisco Platforms Supporting IPv6 Hardware Forwarding

Supported Platforms

[Table 1](#) lists the Cisco platforms that have IPv6 hardware forwarding and the Cisco IOS software release trains that introduce the feature.



Note

[Table 2](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise in [Table 2](#), subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Minimum Required Release for Cisco Platforms Supporting IPv6 Hardware Forwarding

Hardware and Feature	Cisco IOS Software Release
Cisco 12000 Series	
IP ISE line card IPv6 forwarding	12.0(23)S
IP ISE line card extended ACLs	12.0(25)S
IP ISE line card IPv6 over MPLS (6PE)	12.0(25)S
IP ISE line card IPv6 Multicast assist	12.0(26)S
IP ISE line card IPv6 QoS	12.0(28)S
Engine 5 line card IPv6 hardware forwarding	12.0(31)S
IP Receive ACL for IPv6 traffic	12.0(32)S
Cisco 10000 Series	
Cisco 10000 series Performance Routing Engine 2 (PRE-2)	12.2(28)SB
Cisco 10000 series PRE-3	12.2(31)SB
Cisco 10000 series 6PE support	12.2(31)SB
Cisco 10000 series PRE-4	12.2(33)SB
Cisco 10720 Series	
PxF accelerated for IPv6 forwarding	12.0(26)S, 12.2(28)SB
PxF accelerated for IPv6 extended ACLs	12.0(26)S
PxF accelerated for IPv6 over MPLS (6PE)	12.0(26)S
PRE-2 hardware forwarding	12.2(28)SB

Table 2 Minimum Required Release for Cisco Platforms Supporting IPv6 Hardware Forwarding

Hardware and Feature	Cisco IOS Software Release
Cisco 7600 Series, Cisco Catalyst 6500, Cisco Catalyst 3700, and Cisco Catalyst 3500	
IPv6: Express setup	12.2(35)SE
Cisco Catalyst 3560 series	12.2(25)SEA
Cisco Catalyst 3750 series	12.2(25)SEA
IPv6: IPv6 and IPv4 TCAM templates	12.2(25)SEA
IPv6: IPv6 neighbor discovery throttling	12.2(25)SEA
Cisco Catalyst 3560E series	12.2(35)SE2
Cisco Catalyst 3570E series	12.2(35)SE2
Cisco Catalyst 3560 series: IPv6 multicast hardware layer	12.2(25)SED
Supervisor Engines 720 and 720-3bxl	12.2(33)SRA
Route/switch processor 720 on Cisco 7600 series	12.2(33)SRB
Supervisor Engine 720 IPv6 forwarding	12.2(17a)SX1
Supervisor Engine 720 IPv6 extended ACLs	12.2(17a)SX1
Supervisor Engine 720 IPv6 over MPLS (6PE)	12.2(17b)SXA
Supervisor Engine 720 IPv6 multicast hardware forwarding	12.2(18)SXE
Supervisor Engine 720 IPv6 multicast RPR/RPR+ support	12.2(18)SXE
Supervisor Engine 720 IPv6 multicast hardware-assisted egress replication	12.2(18)SXE
Supervisor Engine 32/MSFC2A	12.2(18)SXF
Cisco ASR 1000 Series	
ASR1000 series RP1	Cisco IOS XE Release 2.1

Additional 12.2S Release Trains

Several early-deployment Cisco IOS software Release 12.2S trains synchronize to the Cisco IOS software mainline Release 12.2S train. The following table lists information about the release trains on which IPv6 hardware is used.

Table 3 Minimum Required Release for IPv6 Hardware on Early-Deployment 12.2S Cisco IOS Software Release Trains

Early-Deployment Cisco IOS Software Release and Hardware	Release Description
12.2(28)SB and 12.2(33)SB on Cisco 10000 series	Not all features for Cisco IOS Release 12.2(28)SB or Cisco IOS Release 12.2(33)SB are supported on the Cisco 10000 series routers. For further information on Cisco IOS Release 12.2(28)SB or Cisco IOS Release 12.2(33)SB, see the release notes at the following URLs: http://www.cisco.com/en/US/docs/ios/12_2sb/release/notes/122SB.html
12.2(25)SEA on Cisco Catalyst 3560 and 3570 series	12.2(25)SEA supports a subset of the 12.2S IPv6 feature set. IPv6 multicast is not supported.

Table 3 Minimum Required Release for IPv6 Hardware on Early-Deployment 12.2S Cisco IOS Software Release Trains

Early-Deployment Cisco IOS Software Release and Hardware	Release Description
12.2(33)SRA on Cisco 7600 series	12.2(33)SRA includes all IPv6 features from Cisco IOS software releases 12.2S and 12.2SX.
12.2SX on Cisco Catalyst 6500	12.2(17)SX includes the entire Cisco IOS software Release 12.2(14)S feature set, plus OSPFv3.
12.2(17d)SXB on Cisco Catalyst 6500 Supervisor Engine 2/MSFC2	IPv6 support provided on 12.2(17)SXB for Cisco Catalyst 6500 Supervisor Engine 2/MSFC2.
12.2(18)SXE on Cisco Catalyst 6500 and Cisco 7600 series	12.2(18)SXE supports IPv6 multicast hardware forwarding.
12.2(18)SXF on Supervisor Engine 32/MSFC2A	
12.2(35)SE2 on Cisco Catalyst 3560E and 3570E series	
12.2(40)SE on Cisco Catalyst 2960	IPv6 support provided for MLD snooping.
12.2(33)SCA on UBR	Support is provided for DHCPv6 relay agent notification for prefix delegation.

Additional References

The following sections provide references related to Cisco IOS IPv6 features:

Related Documents

Related Topic	Document Title
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

RFCs

RFCs	Title
RFC 1886	<i>DNS Extensions to Support IP version 6</i>
RFC 1981	<i>Path MTU Discovery for IP version 6</i>
RFC 2080	<i>RIPng for IPv6</i>
RFC 2375	<i>IPv6 Multicast Address Assignments</i>
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2402	<i>IP Authentication Header</i>
RFC 2404	<i>The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header</i>
RFC 2406	<i>IP Encapsulating Security Payload (ESP)</i>

RFCs	Title
RFC 2407	<i>The Internet Security Domain of Interpretation for ISAKMP</i>
RFC 2408	<i>Internet Security Association and Key Management Protocol</i>
RFC 2409	<i>Internet Key Exchange (IKE)</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2461	<i>Neighbor Discovery for IP Version 6 (IPv6)</i>
RFC 2462	<i>IPv6 Stateless Address Autoconfiguration</i>
RFC 2463	<i>Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification</i>
RFC 2464	<i>Transmission of IPv6 Packets over Ethernet</i>
RFC 2467	<i>Transmission of IPv6 Packets over FDDI</i>
RFC 2472	<i>IP Version 6 over PPP</i>
RFC 2474	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>
RFC 2475	<i>An Architecture for Differentiated Services Framework</i>
RFC 2492	<i>IPv6 over ATM</i>
RFC 2545	<i>Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing</i>
RFC 2590	<i>Transmission of IPv6 Packets over Frame Relay Specification</i>
RFC 2597	<i>Assured Forwarding PHB</i>
RFC 2598	<i>An Expedited Forwarding PHB</i>
RFC 2697	<i>A Single Rate Three Color Marker</i>
RFC 2698	<i>A Two Rate Three Color Marker</i>
RFC 2710	<i>Multicast Listener Discovery (MLD) for IPv6</i>
RFC 2711	<i>IPv6 Router Alert Option</i>
RFC 2740	<i>OSPF for IPv6</i>
RFC 2766	<i>Network Address Translation–Protocol Translation (NAT-PT)</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2893	<i>Transition Mechanisms for IPv6 Hosts and Routers</i>
RFC 3056	<i>Connection of IPv6 Domains via IPv4 Clouds</i>
RFC 3068	<i>An Anycast Prefix for 6to4 Relay Routers</i>
RFC 3147	<i>Generic Routing Encapsulation over CLNS</i>
RFC 3162	<i>RADIUS and IPv6</i>
RFC 3315	<i>Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i>
RFC 3319	<i>Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 3484	<i>Default Address Selection for Internet Protocol version 6 (IPv6)</i>
RFC 3513	<i>Internet Protocol Version 6 (IPv6) Addressing Architecture</i>

RFCs	Title
RFC 3576	<i>Change of Authorization</i>
RFC 3587	<i>IPv6 Global Unicast Address Format</i>
RFC 3596	<i>DNS Extensions to Support IP Version 6</i>
RFC 3633	<i>DHCP IPv6 Prefix Delegation</i>
RFC 3646	<i>DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i>
RFC 3736	<i>Stateless DHCP Service for IPv6</i>
RFC 3775	<i>Mobility Support in IPv6</i>
RFC 3810	<i>Multicast Listener Discovery Version 2 (MLDv2) for IPv6</i>
RFC 3879	<i>Deprecating Site Local Addresses</i>
RFC 3954	<i>Cisco Systems NetFlow Services Export Version 9</i>
RFC 3956	<i>Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address</i>
RFC 3963	<i>Network Mobility (NEMO) Basic Support Protocol</i>
RFC 4007	<i>IPv6 Scoped Address Architecture</i>
RFC 4087	<i>IP Tunnel MIB</i>
RFC 4109	<i>Algorithms for Internet Key Exchange version 1 (IKEv1)</i>
RFC 4191	<i>Default Router Preferences and More-Specific Routes</i>
RFC 4193	<i>Unique Local IPv6 Unicast Addresses</i>
RFC 4214	<i>Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)</i>
RFC 4291	<i>IP Version 6 Addressing Architecture</i>
RFC 4292	<i>IP Forwarding Table MIB</i>
RFC 4293	<i>Management Information Base for the Internet Protocol (IP)</i>
RFC 4302	<i>IP Authentication Header</i>
RFC 4306	<i>Internet Key Exchange (IKEv2) Protocol</i>
RFC 4308	<i>Cryptographic Suites for IPsec</i>
RFC 4382	<i>MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base</i>
RFC 4443	<i>Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification</i>
RFC 4552	<i>Authentication/Confidentiality for OSPFv3</i>
RFC 4659	<i>BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN</i>
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>
RFC 4798	<i>Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)</i>
RFC 5095	<i>Deprecation of Type 0 Routing Headers in IPv6</i>

RFCs	Title
RFC 5120	<i>M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)</i>
RFC 5308	<i>Routing IPv6 with IS-IS</i>

The draft RFCs supported are as follows:

- draft-bonica-internet-icmp, *ICMP Extensions for Multiprotocol Label Switching*
- draft-ietf-pim-sm-v2-new, *Protocol Independent Multicast - Sparse Mode PIM-SM: Protocol Specification (Revised)*, March 6, 2003
- draft-ietf-pim-sm-bsr-03.txt, *Bootstrap Router (BSR) Mechanism for PIM Sparse Mode*, February 25, 2003
- draft-suz-pim-upstream-detection, *PIM Upstream Detection Among Multiple Addresses*, February 2003

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> • CISCO-CONFIG-COPY-MIB • CISCO-CONFIG-MAN-MIB • CISCO-DATA-COLLECTION-MIB • CISCO-FLASH-MIB • CISCO-IETF-IP-FORWARDING-MIB (not available as of Cisco IOS Release 12.2(33)SRC) • CISCO-IETF-IP-MIB (not available as of Cisco IOS Release 12.2(33)SRC) • CISCO-IP-FORWARD-MIB • CISCO-IP-MIB • CISCO-SNMP-TARGET-EXT-MIB • ENTITY-MIB • NOTIFICATION-LOG-MIB • SNMP-TARGET-MIB • TUNNEL-MIB 	<p>To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL:</p> <p>http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</p>

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 Addressing and Basic Connectivity

First Published: June 7, 2001

Last Updated: August 18, 2008

Implementing basic IPv6 connectivity in the Cisco IOS software consists of assigning IPv6 addresses to individual router interfaces. The forwarding of IPv6 traffic can be enabled globally, and Cisco Express Forwarding switching for IPv6 can also be enabled. Basic connectivity can be enhanced by configuring support for AAAA record types in the Domain Name System (DNS) name-to-address and address-to-name lookup processes, and by managing IPv6 neighbor discovery.

This module describes IPv6 addressing and basic IPv6 connectivity tasks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPv6 Addressing and Basic Connectivity” section on page 62](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IPv6 Addressing and Basic Connectivity, page 2](#)
- [Restrictions for Implementing IPv6 Addressing and Basic Connectivity, page 2](#)
- [Information About Implementing IPv6 Addressing and Basic Connectivity, page 3](#)
- [How to Implement IPv6 Addressing and Basic Connectivity, page 27](#)
- [Configuration Examples for Implementing IPv6 Addressing and Basic Connectivity, page 51](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006–2009 Cisco Systems, Inc. All rights reserved.

- [Where to Go Next, page 56](#)
- [Additional References, page 57](#)
- [Command Reference, page 59](#)
- [Feature Information for Implementing IPv6 Addressing and Basic Connectivity, page 62](#)

Prerequisites for Implementing IPv6 Addressing and Basic Connectivity

- This document assumes that you are familiar with IPv4. See the publications shown in the [“Additional References”](#) section for IPv4 configuration and command reference information.
- The following prerequisites apply to Cisco Express Forwarding and distributed Cisco Express Forwarding for IPv6:
 - To forward IPv6 traffic using Cisco Express Forwarding or distributed Cisco Express Forwarding, you must configure forwarding of IPv6 unicast datagrams globally on the router by using the **ipv6 unicast-routing** command, and you must configure an IPv6 address on an interface by using the **ipv6 address** command.
 - You must enable Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef** command before enabling Cisco Express Forwarding for IPv6 globally on the router by using the **ipv6 cef** command.
 - On distributed architecture platforms that support both Cisco Express Forwarding and distributed Cisco Express Forwarding, such as the Cisco 7500 series routers, you must enable distributed Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef distributed** command before enabling distributed Cisco Express Forwarding for IPv6 globally on the router by using the **ipv6 cef distributed** command.



Note

By default, the Cisco 12000 series Internet routers support only distributed Cisco Express Forwarding.

- To use Unicast Reverse Path Forwarding (RPF), enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the router. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the router, individual interfaces can be configured with other switching modes.



Note

For Unicast RPF to work, Cisco Express Forwarding must be configured globally in the router. Unicast RPF will not work without Cisco Express Forwarding.

Restrictions for Implementing IPv6 Addressing and Basic Connectivity

- In Cisco IOS Release 12.2(11)T or earlier releases, IPv6 supports only process switching for packet forwarding. Cisco Express Forwarding switching and distributed Cisco Express Forwarding switching for IPv6 are supported in Cisco IOS Release 12.2(13)T. Distributed Cisco Express Forwarding switching for IPv6 is supported in Cisco IOS Release 12.0(21)ST.

- IPv6 packets are transparent to Layer 2 LAN switches because the switches do not examine Layer 3 packet information before forwarding IPv6 frames. Therefore, IPv6 hosts can be directly attached to Layer 2 LAN switches.
- In any Cisco IOS release with IPv6 support, multiple IPv6 global addresses within the same prefix can be configured on an interface. However, multiple IPv6 link-local addresses on an interface are not supported. See the “[IPv6 Addressing and IPv6 Routing Configuration: Example](#)” section for information on configuring multiple IPv6 global addresses within the same prefix on an interface.
- Because RFC 3879 deprecates the use of site-local addresses, configuration of private IPv6 addresses should be done following the recommendations of unique local addressing (ULA) in RFC 4193.

Information About Implementing IPv6 Addressing and Basic Connectivity

To configure IPv6 addressing and basic connectivity for IPv6 for Cisco IOS software, you must understand the following concepts:

- [IPv6 for Cisco IOS Software, page 4](#)
- [Large IPv6 Address Space for Unique Addresses, page 4](#)
- [IPv6 Address Formats, page 4](#)
- [IPv6 Address Type: Unicast, page 5](#)
- [IPv6 Address Type: Anycast, page 9](#)
- [IPv6 Address Type: Multicast, page 10](#)
- [IPv6 Address Output Display, page 11](#)
- [Simplified IPv6 Packet Header, page 12](#)
- [Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 15](#)
- [DNS for IPv6, page 17](#)
- [Path MTU Discovery for IPv6, page 17](#)
- [Cisco Discovery Protocol IPv6 Address Support, page 17](#)
- [ICMP for IPv6, page 18](#)
- [IPv6 Neighbor Discovery, page 18](#)
- [Link, Subnet, and Site Addressing Changes, page 24](#)
- [IPv6 Prefix Aggregation, page 25](#)
- [IPv6 Site Multihoming, page 26](#)
- [IPv6 Data Links, page 26](#)
- [Routed Bridge Encapsulation for IPv6, page 26](#)
- [Dual IPv4 and IPv6 Protocol Stacks, page 26](#)

IPv6 for Cisco IOS Software

IPv6, formerly named IPng (next generation), is the latest version of the Internet Protocol (IP). IP is a packet-based protocol used to exchange data, voice, and video traffic over digital networks. IPv6 was proposed when it became clear that the 32-bit addressing scheme of IP version 4 (IPv4) was inadequate to meet the demands of Internet growth. After extensive discussion it was decided to base IPng on IP but add a much larger address space and improvements such as a simplified main header and extension headers. IPv6 is described initially in RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*, issued by the Internet Engineering Task Force (IETF). Further RFCs describe the architecture and services supported by IPv6.

The architecture of IPv6 has been designed to allow existing IPv4 users to transition easily to IPv6 while providing services such as end-to-end security, quality of service (QoS), and globally unique addresses. The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. IPv6 prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities provide an IPv6 addressing hierarchy that allows for more efficient routing. IPv6 supports widely deployed routing protocols such as Routing Information Protocol (RIP), Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First for IPv6, and multiprotocol Border Gateway Protocol (BGP). Other available features include stateless autoconfiguration, enhanced support for Mobile IPv6, and an increased number of multicast addresses.

Large IPv6 Address Space for Unique Addresses

The primary motivation for IPv6 is the need to meet the anticipated future demand for globally unique IP addresses. Applications such as mobile Internet-enabled devices (such as personal digital assistants [PDAs], telephones, and cars), home-area networks (HANs), and wireless data services are driving the demand for globally unique IP addresses. IPv6 quadruples the number of network address bits from 32 bits (in IPv4) to 128 bits, which provides more than enough globally unique IP addresses for every networked device on the planet. By being globally unique, IPv6 addresses inherently enable global reachability and end-to-end security for networked devices, functionality that is crucial to the applications and services that are driving the demand for the addresses. Additionally, the flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT); therefore, IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 Address Formats

IPv6 addresses are represented as a series of 16-bit hexadecimal fields separated by colons (:) in the format: x:x:x:x:x:x:x:x. Following are two examples of IPv6 addresses:

2001:0DB8:7654:3210:FEDC:BA98:7654:3210

2001:0DB8:0:0:8:800:200C:417A

It is common for IPv6 addresses to contain successive hexadecimal fields of zeros. To make IPv6 addresses less cumbersome, two colons (::) may be used to compress successive hexadecimal fields of zeros at the beginning, middle, or end of an IPv6 address (the colons represent successive hexadecimal fields of zeros). [Table 1](#) lists compressed IPv6 address formats.

A double colon may be used as part of the *ipv6-address* argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.

**Note**

Two colons (::) can be used only once in an IPv6 address to represent the longest successive hexadecimal fields of zeros.

The hexadecimal letters in IPv6 addresses are not case-sensitive.

Table 1 *Compressed IPv6 Address Formats*

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:0DB8:800:200C:417A	2001::0DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0	::

The loopback address listed in [Table 1](#) may be used by a node to send an IPv6 packet to itself. The loopback address in IPv6 functions the same as the loopback address in IPv4 (127.0.0.1).

**Note**

The IPv6 loopback address cannot be assigned to a physical interface. A packet that has the IPv6 loopback address as its source or destination address must remain within the node that created the packet. IPv6 routers do not forward packets that have the IPv6 loopback address as their source or destination address.

The unspecified address listed in [Table 1](#) indicates the absence of an IPv6 address. For example, a newly initialized node on an IPv6 network may use the unspecified address as the source address in its packets until it receives its IPv6 address.

**Note**

The IPv6 unspecified address cannot be assigned to an interface. The unspecified IPv6 addresses must not be used as destination addresses in IPv6 packets or the IPv6 routing header.

An IPv6 address prefix, in the format *ipv6-prefix/prefix-length*, can be used to represent bit-wise contiguous blocks of the entire address space. The *ipv6-prefix* must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

IPv6 Address Type: Unicast

An IPv6 unicast address is an identifier for a single interface, on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address. The Cisco IOS software supports the following IPv6 unicast address types:

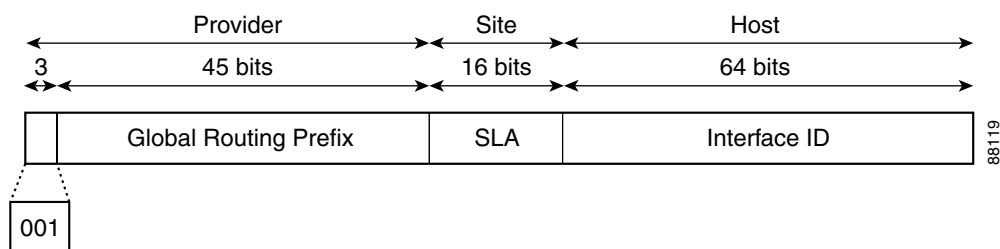
- [Aggregatable Global Address, page 6](#)
- [Link-Local Address, page 7](#)
- [IPv4-Compatible IPv6 Address, page 8](#)
- [Unique Local Address, page 8](#)

Aggregatable Global Address

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses are used on links that are aggregated upward through organizations, and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). [Figure 1](#) shows the structure of an aggregatable global address.

Figure 1 Aggregatable Global Address Format



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address typically consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields named Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLA and NLA fields from the RFCs because these fields are policy-based. Some existing IPv6 networks deployed before the change might still be using networks based on the older architecture.

A 16-bit subnet field called the subnet ID could be used by individual organizations to create their own local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID is used to identify interfaces on a link. The interface ID must be unique to the link. It may also be unique over a broader scope. In many cases, an interface ID will be the same as or based on the link-layer address of an interface. Interface IDs used in aggregatable global unicast and other IPv6 address types must be 64 bits long and constructed in the modified EUI-64 format.

Interface IDs are constructed in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and FDDI interfaces), the first three octets (24 bits) are taken from the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (the Media Access Control [MAC] address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are taken from the last three octets of the MAC address. The construction of the interface ID is completed by setting the Universal/Local (U/L) bit—the seventh bit of the first octet—to a value of 0 or 1. A value of 0 indicates a locally administered identifier; a value of 1 indicates a globally unique IPv6 interface identifier.

- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel interface types—except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is constructed in the same way as the interface ID for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used to construct the identifier (because the interface does not have a MAC address).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface ID is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier.

**Note**

For interfaces using Point-to-Point Protocol (PPP), given that the interfaces at both ends of the connection might have the same MAC address, the interface identifiers used at both ends of the connection are negotiated (picked randomly and, if necessary, reconstructed) until both identifiers are unique. The first MAC address in the router is used to construct the identifier for interfaces using PPP.

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

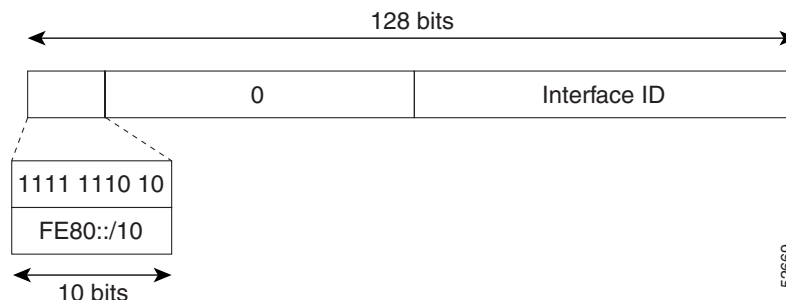
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest algorithm 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

Link-Local Address

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the neighbor discovery protocol and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need globally unique addresses to communicate. [Figure 2](#) shows the structure of a link-local address.

IPv6 routers must not forward packets that have link-local source or destination addresses to other links.

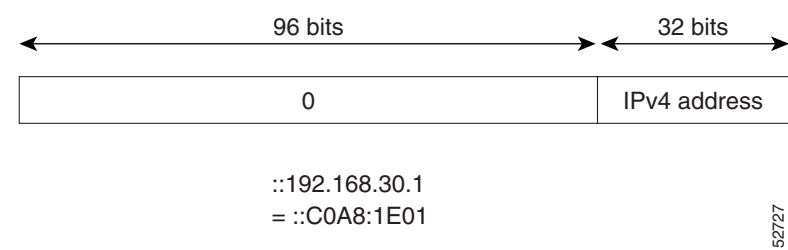
Figure 2 *Link-Local Address Format*



IPv4-Compatible IPv6 Address

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32 bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels. Figure 3 shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 3 IPv4-Compatible IPv6 Address Format



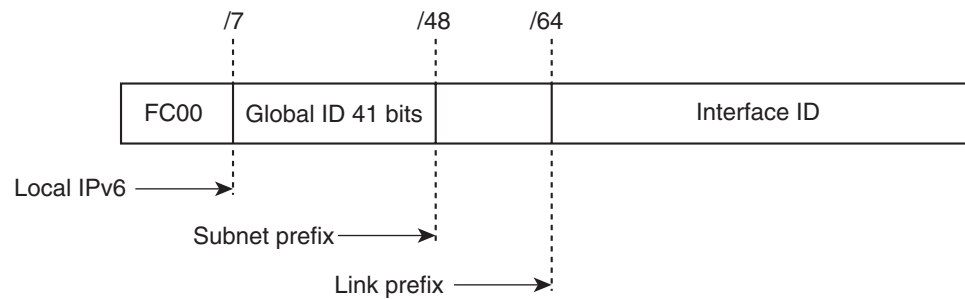
Unique Local Address

A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. They are not expected to be routable on the global Internet and are routable inside of a limited area, such as a site. They may also be routed between a limited set of sites.

A unique local address has the following characteristics:

- It has a globally unique prefix (that is, it has a high probability of uniqueness).
- It has a well-known prefix to allow for easy filtering at site boundaries.
- It allows sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- It is ISP-independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- If it is accidentally leaked outside of a site via routing or DNS, there is no conflict with any other addresses.
- Applications may treat unique local addresses like global scoped addresses.

Figure 4 shows the structure of a unique local address.

Figure 4 Unique Local Address Structure

- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit IID

232389

Site-Local Address

Because RFC 3879 deprecates the use of site-local addresses, configuration of private IPv6 addresses should be done following the recommendations of unique local addressing (ULA) in RFC 4193.

IPv6 Address Type: Anycast

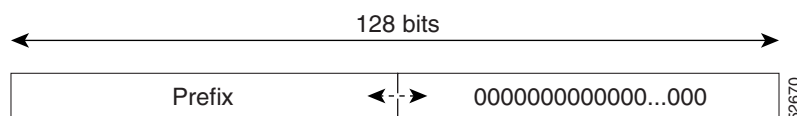
An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface makes a unicast address an anycast address. Nodes to which the anycast address is assigned must be explicitly configured to recognize that the address is an anycast address.



Note

Anycast addresses can be used only by a router, not a host, and anycast addresses must not be used as the source address of an IPv6 packet.

Figure 5 shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

Figure 5 Subnet Router Anycast Address Format

The following shows the configuration for an anycast prefix for 6to4 relay routers:

```
interface Tunnel0
no ip address
```

```

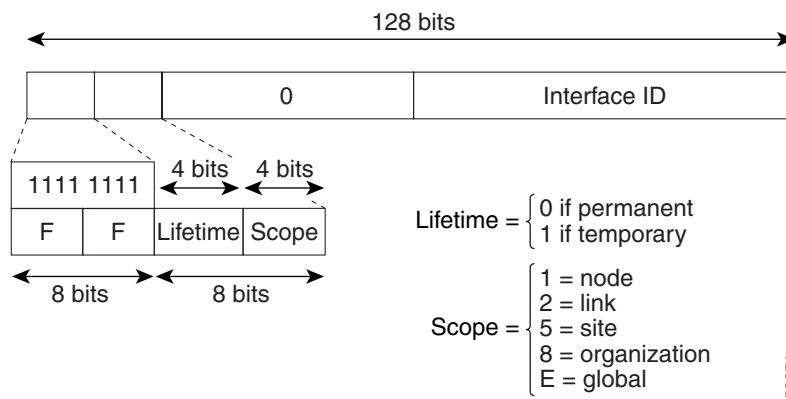
ipv6 address 2001:0DB8:A00:1::1/64
ipv6 address 2001:0DB8:c058:6301::/128 anycast
tunnel source Ethernet0
tunnel mode ipv6ip 6to4
!
interface Ethernet0
ip address 10.0.0.1 255.255.255.0
ip address 192.88.99.1 255.255.255.0 secondary
!
ipv6 route 2001:0DB8::/16 Tunnel0
!

```

IPv6 Address Type: Multicast

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. [Figure 6](#) shows the format of the IPv6 multicast address.

Figure 6 IPv6 Multicast Address Format

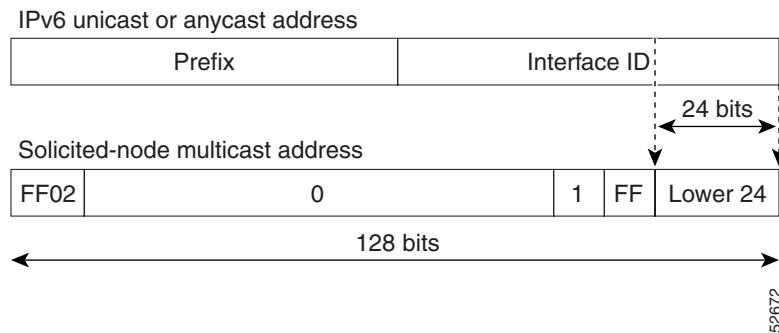


IPv6 nodes (hosts and routers) are required to join (receive packets destined for) the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:0:1 (scope is link-local)
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address (see [Figure 7](#)). For example, the solicited-node multicast address corresponding to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

Figure 7 IPv6 Solicited-Node Multicast Address Format**Note**

There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

For further information on IPv6 multicast, see the [Implementing IPv6 Multicast](#) document.

IPv6 Address Output Display

When IPv6 or IPv4 command output displays an IPv6 address, a long IPv6 address can overflow into neighboring fields, causing the output to be difficult to read. The output fields were designed to work with the longest possible IPv4 address, which has 15 characters; IPv6 addresses can be up to 39 characters long. The following scheme has been adopted in IPv4 and IPv6 commands to allow the appropriate length of IPv6 address to be displayed and move the following fields to the next line, if necessary. The fields that are moved are kept in alignment with the header row.

Using the output display from the **where** command as an example, eight connections are displayed. The first six connections feature IPv6 addresses; the last two connections feature IPv4 addresses.

Router# **where**

Conn	Host	Address	Byte	Idle	Conn	Name
1	test5	2001:0DB8:3333:4::5	6	24	test5	
2	test4	2001:0DB8:3333:44::5	6	24	test4	
3	2001:0DB8:3333:4::5	2001:0DB8:3333:4::5	6	24	2001:0DB8:3333:4::5	
4	2001:0DB8:3333:44::5	2001:0DB8:3333:44::5	6	23	2001:0DB8:3333:44::5	
5	2001:0DB8:3000:4000:5000:6000:7000:8001	2001:0DB8:3000:4000:5000:6000:7000:8001	6	20	2001:0DB8:3000:4000:5000:6000:	
6	2001:0DB8:1::1	2001:0DB8:1::1	0	1	2001:0DB8:1::1	
7	10.1.9.1	10.1.9.1	0	0	10.1.9.1	
8	10.222.111.222	10.222.111.222	0	0	10.222.111.222	

Connection 1 contains an IPv6 address that uses the maximum address length in the address field. Connection 2 shows the IPv6 address overflowing the address field and the following fields moved to the next line, but in alignment with the appropriate headers. Connection 3 contains an IPv6 address that fills the maximum length of the hostname and address fields without wrapping any lines. Connection 4 shows the effect of both the hostname and address fields containing a long IPv6 address. The output is shown over three lines keeping the correct heading alignment. Connection 5 displays a similar effect as

connection 4 with a very long IPv6 address in the hostname and address fields. Note that the connection name field is actually truncated. Connection 6 displays a very short IPv6 address that does not require any change in the display. Connections 7 and 8 display short and long IPv4 addresses.

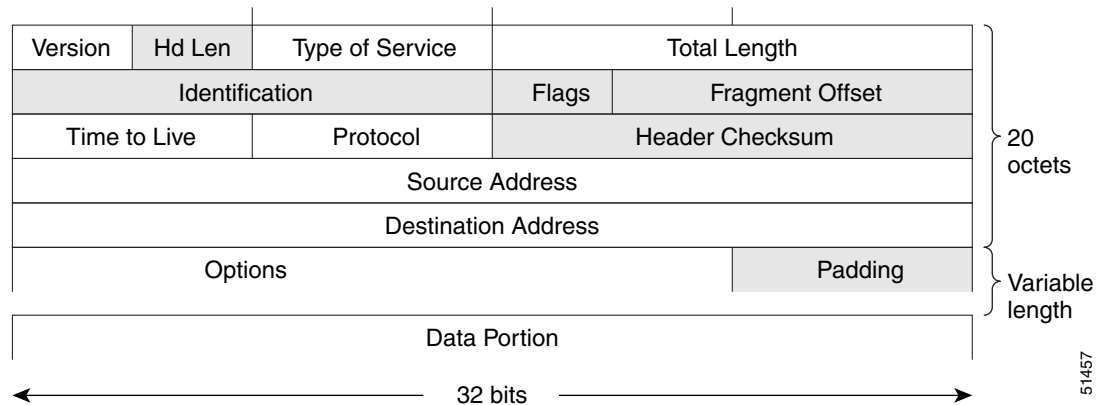
**Note**

The IPv6 address output display applies to all commands that display IPv6 addresses.

Simplified IPv6 Packet Header

The basic IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see [Figure 8](#)). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header shown in [Figure 8](#) are not included in the IPv6 packet header.

Figure 8 IPv4 Packet Header Format



The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see [Figure 9](#)). Fields were removed from the IPv6 header because, in IPv6, fragmentation is not handled by routers and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. (In IPv4, the User Datagram Protocol (UDP) transport layer uses an optional checksum. In IPv6, use of the UDP checksum is required to check the integrity of the inner packet.) Additionally, the basic IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

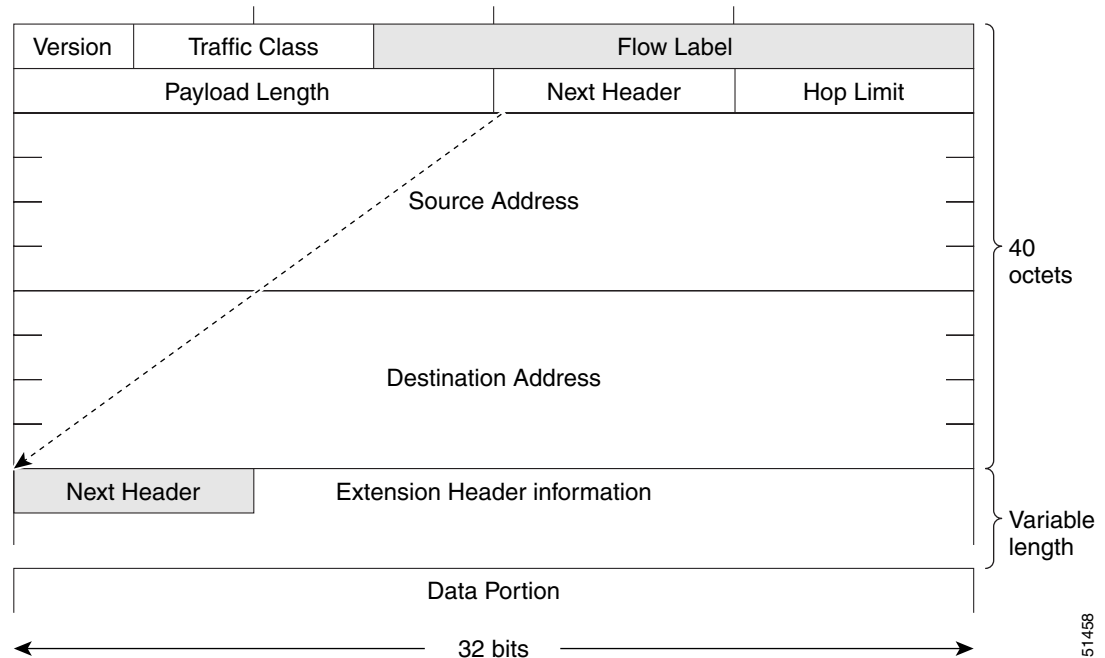
Figure 9 **IPv6 Packet Header Format**

Table 2 lists the fields in the basic IPv6 packet header.

Table 2 **Basic IPv6 Packet Header Fields**

Field	Description
Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	A new field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information following the basic IPv6 header. The type of information following the basic IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header, as shown in Figure 9.
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.

Table 2 Basic IPv6 Packet Header Fields (continued)

Field	Description
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Following the eight fields of the basic IPv6 packet header are optional extension headers and the data portion of the packet. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Together, the extension headers form a chain of headers. Each extension header is identified by the Next Header field of the previous header. Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP. [Figure 10](#) shows the IPv6 extension header format.

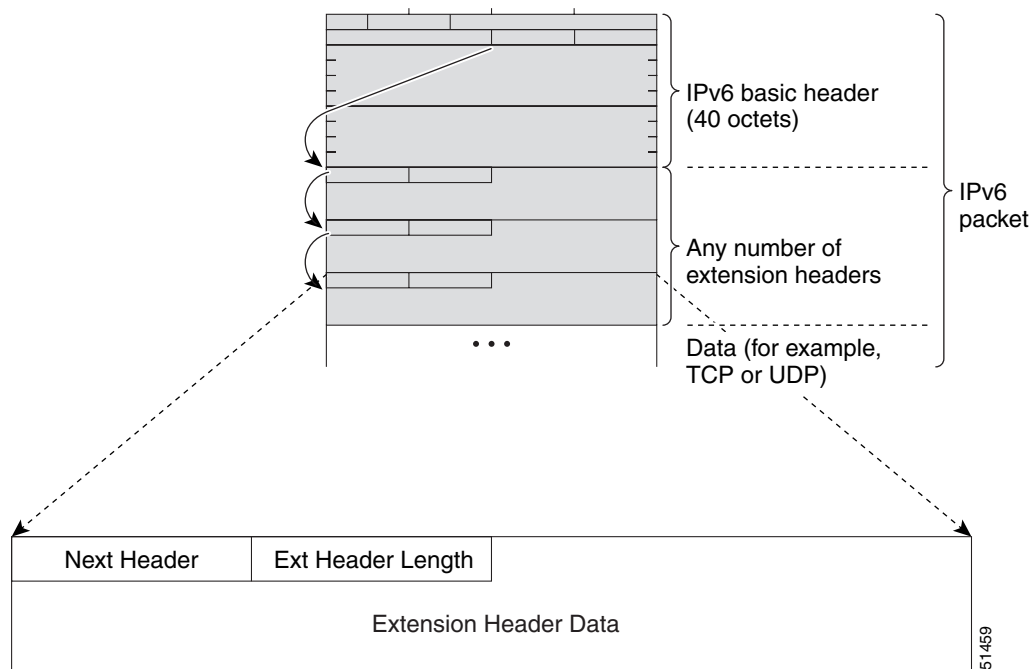
Figure 10 IPv6 Extension Header Format

Table 3 lists the extension header types and their Next Header field values.

Table 3 *IPv6 Extension Header Types*

Header Type	Next Header Value	Description
Hop-by-hop options header	0	This header is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.
Destination options header	60	The destination options header can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination.
Routing header	43	The routing header is used for source routing.
Fragment header	44	The fragment header is used when a source must fragment a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Authentication header and ESP header	51 50	The Authentication header and the ESP header are used within IP Security Protocol (IPsec) to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.
Upper-layer headers	6 (TCP) 17 (UDP)	The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP and UDP.
Mobility headers	135	Extension headers used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings.

Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6

Cisco Express Forwarding is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets. Distributed Cisco Express Forwarding performs the same functions as Cisco Express Forwarding but for distributed architecture platforms such as the Cisco 12000 series Internet routers and the Cisco 7500 series routers. Distributed Cisco Express Forwarding for IPv6 and Cisco Express Forwarding for IPv6 function the same and offer the same benefits as for distributed Cisco Express Forwarding for IPv4 and Cisco Express Forwarding for IPv4—network entries that are added, removed, or modified in the IPv6 Routing Information Base (RIB), as dictated by the routing protocols in use, are reflected in the Forwarding Information Bases (FIBs), and the IPv6 adjacency tables maintain Layer 2 next-hop addresses for all entries in each FIB.

**Note**

By default, the Cisco 12000 series Internet routers support only distributed Cisco Express Forwarding (Cisco Express Forwarding switching is performed by the line cards). The Cisco 7500 series routers support both Cisco Express Forwarding and distributed Cisco Express Forwarding. When Cisco Express Forwarding is configured on Cisco 7500 series routers, Cisco Express Forwarding switching is performed by the Route Processor (RP); when distributed Cisco Express Forwarding is configured, Cisco Express Forwarding switching is performed by the line cards.

In Cisco IOS Release 12.0(21)ST, distributed Cisco Express Forwarding included support for IPv6 addresses and prefixes. In Cisco IOS Release 12.0(22)S or later releases and Cisco IOS Release 12.2(13)T or later releases, distributed Cisco Express Forwarding and Cisco Express Forwarding were enhanced to include support for separate FIBs for IPv6 global and link-local addresses.

Each IPv6 router interface has an association to one IPv6 global FIB and one IPv6 link-local FIB (multiple interfaces can have an association to the same FIB). All IPv6 router interfaces that are attached to the same IPv6 link share the same IPv6 link-local FIB. IPv6 packets that have an IPv6 global destination address are processed by the IPv6 global FIB; however, packets that have an IPv6 global destination address and an IPv6 link-local source address are sent to the RP for process switching and scope-error handling. Packets that have a link-local source address are not forwarded off of the local link and are sent to the RP for process switching and scope-error handling.

Unicast Reverse Path Forwarding

Use the Unicast RPF feature to mitigate problems caused by malformed or forged (spoofed) IPv6 source addresses that pass through an IPv6 router. Malformed or forged source addresses can indicate denial-of-service (DoS) attacks based on source IPv6 address spoofing.

When Unicast RPF is enabled on an interface, the router examines all packets received on that interface. The router verifies that the source address appears in the routing table and matches the interface on which the packet was received. This “look backward” ability is available only when Cisco Express Forwarding is enabled on the router, because the lookup relies on the presence of the FIB. Cisco Express Forwarding generates the FIB as part of its operation.

**Note**

Unicast RPF is an input function and is applied only on the input interface of a router at the upstream end of a connection.

The Unicast RPF feature verifies whether any packet received at a router interface arrives on one of the best return paths to the source of the packet. The feature performs a reverse lookup in the Cisco Express Forwarding table. If Unicast RPF does not find a reverse path for the packet, Unicast RPF can drop or forward the packet, depending on whether an access control list (ACL) is specified. If an ACL is specified, then when (and only when) a packet fails the Unicast RPF check, the ACL is checked to verify if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL). Whether a packet is dropped or forwarded, the packet is counted in the global IP traffic statistics for Unicast RPF drops and in the interface statistics for Unicast RPF.

If no ACL is specified, the router drops the forged or malformed packet immediately and no ACL logging occurs. The router and interface Unicast RPF counters are updated.

Unicast RPF events can be logged by specifying the logging option for the ACL entries. Log information can be used to gather information about the attack, such as source address and time.

**Note**

With Unicast RPF, all equal-cost “best” return paths are considered valid. Unicast RPF works in cases where multiple return paths exist, provided that each path is equal to the others in terms of the routing cost (number of hops, weights, and so on) and as long as the route is in the FIB.

DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses. IPv6 also supports the reverse mapping of IPv6 addresses to DNS names.

**Note**

IP6.ARPA support was added in the Cisco IOS 12.3(11)T release. IP6.ARPA is not supported in releases prior to the Cisco IOS 12.3(11)T release.

Table 4 lists the IPv6 DNS record types.

Table 4 **IPv6 DNS Record Types**

Record Type	Description	Format
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.) Note Support for AAAA records and A records over an IPv6 transport or IPv4 transport is in Cisco IOS Release 12.2(8)T or later releases.	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.) Note The Cisco IOS software supports resolution of PTR records for the IP6.INT domain.	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.1.c.0.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

Path MTU Discovery for IPv6

As in IPv4, path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently.

**Note**

In IPv6, the minimum link MTU is 1280 octets. Cisco recommends using an MTU value of 1500 octets for IPv6 links.

Cisco Discovery Protocol IPv6 Address Support

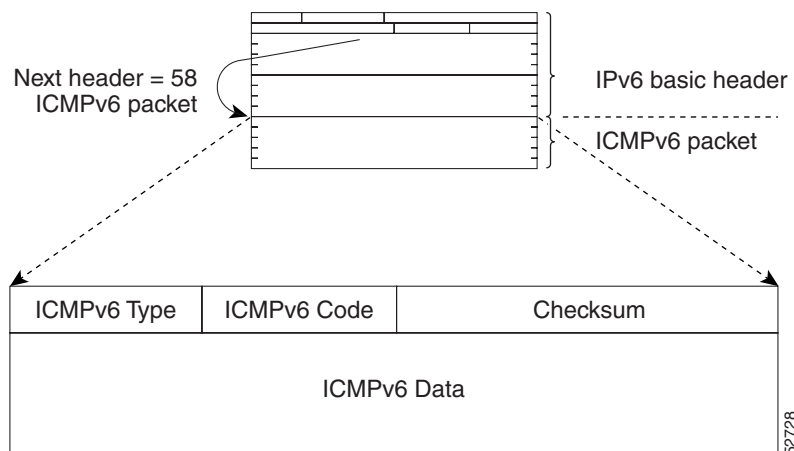
The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices. Cisco Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

ICMP for IPv6

Internet Control Message Protocol (ICMP) in IPv6 functions the same as ICMP in IPv4. ICMP generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the Multicast Listener Discovery (MLD) protocol for IPv6. MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. MLD is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4.

A value of 58 in the Next Header field of the basic IPv6 packet header identifies an IPv6 ICMP packet. ICMP packets in IPv6 are like a transport-layer packet in the sense that the ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet. Within IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is derived (computed by the sender and checked by the receiver) from the fields in the IPv6 ICMP packet and the IPv6 pseudoheader. The ICMPv6 Data field contains error or diagnostic information relevant to IP packet processing. [Figure 11](#) shows the IPv6 ICMP packet header format.

Figure 11 IPv6 ICMP Packet Header Format



IPv6 Neighbor Discovery

The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring routers.

The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache. Static routing requires an administrator to manually enter IPv6 addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each router into a table. Static routing enables more control but requires more work to maintain the table. The table must be updated each time routes are added or changed.

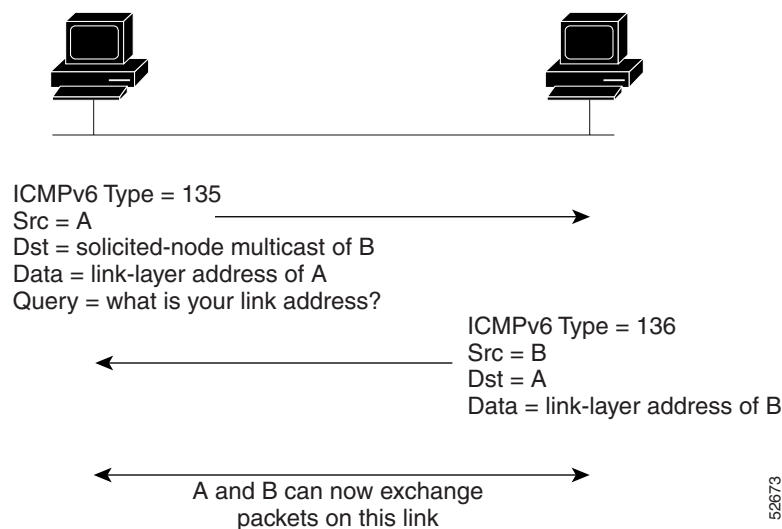
Stateful Switchover

IPv6 neighbor discovery supports stateful switchover (SSO) using Cisco Express Forwarding. When switchover occurs, the Cisco Express Forwarding adjacency state, which is checkpointed, is used to reconstruct the neighbor discovery cache.

IPv6 Neighbor Solicitation Message

A value of 135 in the Type field of the ICMP packet header identifies a neighbor solicitation message. Neighbor solicitation messages are sent on the local link when a node wants to determine the link-layer address of another node on the same local link (see [Figure 12](#)). When a node wants to determine the link-layer address of another node, the source address in a neighbor solicitation message is the IPv6 address of the node sending the neighbor solicitation message. The destination address in the neighbor solicitation message is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 12 IPv6 Neighbor Discovery—Neighbor Solicitation Message



After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address in the neighbor advertisement message is the IPv6 address of the node (more specifically, the IPv6 address of the node interface) sending the neighbor advertisement message. The destination address in the neighbor advertisement message is the IPv6 address of the node that sent the neighbor solicitation message. The data portion of the neighbor advertisement message includes the link-layer address of the node sending the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. When a node wants to verify the reachability of a neighbor, the destination address in a neighbor solicitation message is the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is such a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor solicitation messages are also used to verify the reachability of a neighbor after the link-layer address of a neighbor is identified. Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor, and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive acknowledgment—from an upper-layer protocol (such as TCP)—indicates that a connection is making forward progress (reaching its destination) or the receipt of a neighbor advertisement message in response to a neighbor solicitation message. If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Therefore, forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working.

The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

**Note**

A neighbor advertisement message that has the solicited flag set to a value of 0 must not be considered as a positive acknowledgment that the forward path is still working.

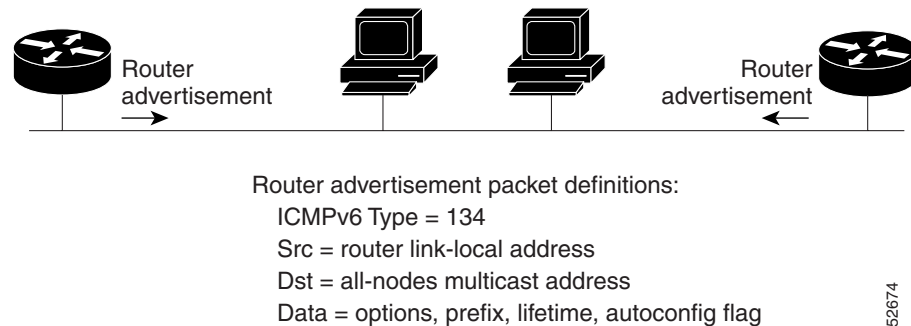
Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). Specifically, a node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

Every IPv6 unicast address (global or link-local) must be verified for uniqueness on the link; however, until the uniqueness of the link-local address is verified, duplicate address detection is not performed on any other IPv6 addresses associated with the link-local address. The Cisco implementation of duplicate address detection in the Cisco IOS software does not verify the uniqueness of anycast or global addresses that are generated from 64-bit interface identifiers.

IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out each configured interface of an IPv6 router. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits. The RA messages are sent to the all-nodes multicast address (see [Figure 13](#)).

Figure 13 IPv6 Neighbor Discovery—RA Message



RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Lifetime information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed
- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time (in seconds) the router should be used as a default router)
- Additional information for hosts, such as the hop limit and MTU a host should use in packets that it originates

RAs are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message. Given that router solicitation messages are usually sent by hosts at system startup (the host does not have a configured unicast address), the source address in router solicitation messages is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface sending the router solicitation message is used as the source address in the message. The destination address in router solicitation messages is the all-routers multicast address with a scope of the link. When an RA is sent in response to a router solicitation, the destination address in the RA message is the unicast address of the source of the router solicitation message.

The following RA message parameters can be configured:

- The time interval between periodic RA messages
- The “router lifetime” value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on Ethernet and FDDI interfaces when the **ipv6 unicast-routing** command is configured. For other interface types, the sending of RA messages must be manually configured by using the **no ipv6 nd ra suppress** command. The sending of RA messages can be disabled on individual interfaces by using the **ipv6 nd ra suppress** command.

Default Router Preferences for Traffic Engineering

Hosts discover and select default routers by listening to RAs. Typical default router selection mechanisms are suboptimal in certain cases, such as when traffic engineering is needed. For example, two routers on a link may provide equivalent but not equal-cost routing, and policy may dictate that one of the routers is preferred. Some examples are as follows:

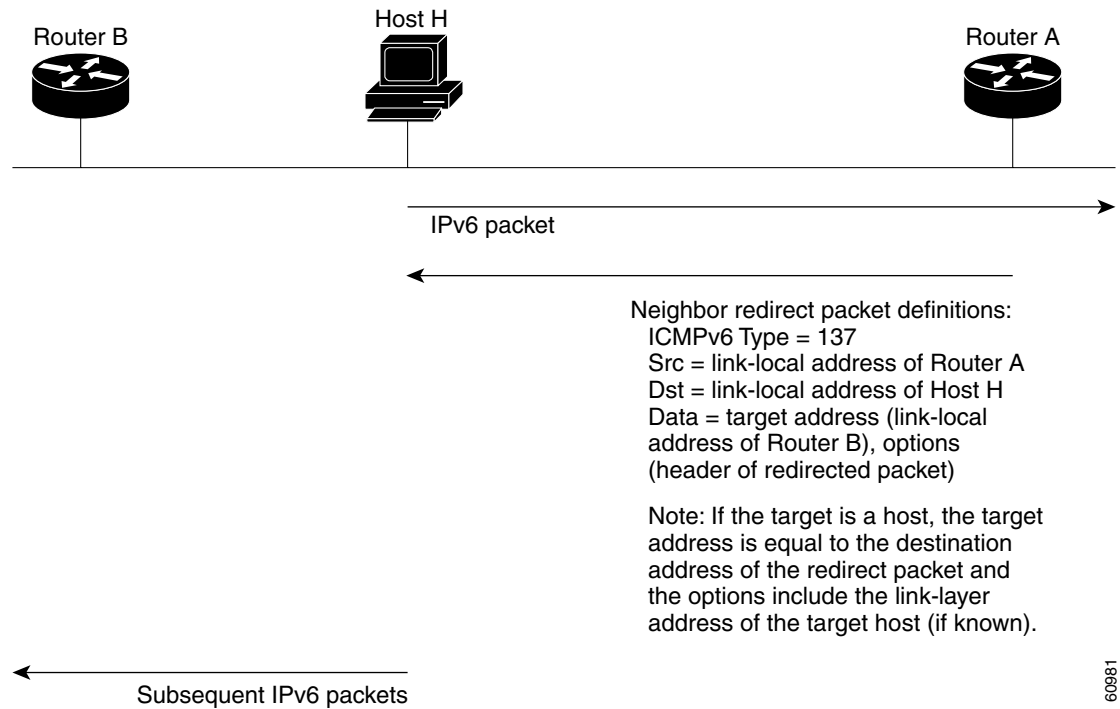
- Multiple routers that route to distinct sets of prefixes—Redirects (sent by nonoptimal routers for a destination) mean that hosts can choose any router and the system will work. However, traffic patterns may mean that choosing one of the routers would lead to considerably fewer redirects.
- Accidentally deploying a new router—Deploying a new router before it has been fully configured could lead to hosts adopting the new router as a default router and traffic disappearing. Network managers may want to indicate that some routers are more preferred than others.
- Multihomed situations—Multihomed situations may become more common, because of multiple physical links and because of the use of tunneling for IPv6 transport. Some of the routers may not provide full default routing because they route only to the 6-to-4 prefix or they route only to a corporate intranet. These situations cannot be resolved with redirects, which operate only over a single link.

The default router preference (DRP) extension provides a coarse preference metric (low, medium, or high) for default routers. The DRP of a default router is signaled in unused bits in RA messages. This extension is backward compatible, both for routers (setting the DRP bits) and hosts (interpreting the DRP bits). These bits are ignored by hosts that do not implement the DRP extension. Similarly, the values sent by routers that do not implement the DRP extension will be interpreted by hosts that do implement it as indicating a “medium” preference.

DRPs need to be configured manually. For information on configuring the optional DRP extension, see the “[Configuring the DRP Extension for Traffic Engineering](#)” section.

IPv6 Neighbor Redirect Message

A value of 137 in the type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination (see [Figure 14](#)).

Figure 14 IPv6 Neighbor Discovery—Neighbor Redirect Message**Note**

A router must be able to determine the link-local address for each of its neighboring routers in order to ensure that the target address (the final destination) in a redirect message identifies the neighbor router by its link-local address. For static routing, the address of the next-hop router should be specified using the link-local address of the router; for dynamic routing, all IPv6 routing protocols must exchange the link-local addresses of neighboring routers.

After forwarding a packet, a router should send a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the router.
- The packet is about to be sent out the interface on which it was received.
- The router determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link, or a link-local address.

Use the **ipv6 icmp error-interval** command to limit the rate at which the router generates all IPv6 ICMP error messages, including neighbor redirect messages, which ultimately reduces link-layer congestion.

**Note**

A router must not update its routing tables after receiving a neighbor redirect message, and hosts must not originate neighbor redirect messages.

Link, Subnet, and Site Addressing Changes

This section describes the IPv6 stateless autoconfiguration and general prefix features, which can be used to manage link, subnet, and site addressing changes.

IPv6 Stateless Autoconfiguration

All interfaces on IPv6 nodes must have a link-local address, which is usually automatically configured from the identifier for an interface and the link-local prefix FE80::/10. A link-local address enables a node to communicate with other nodes on the link and can be used to further configure the node.

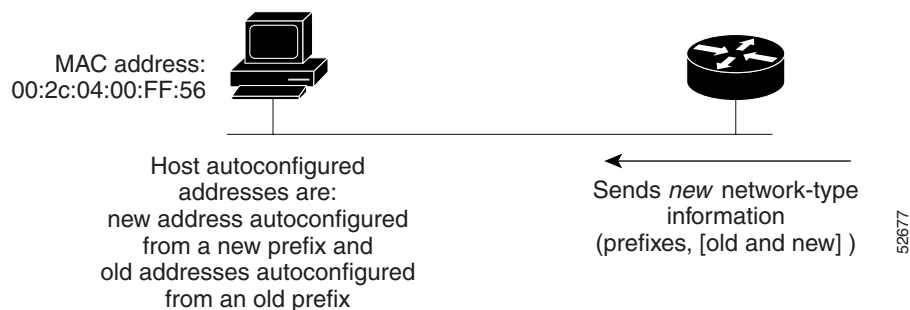
Nodes can connect to a network and automatically generate global IPv6 addresses without the need for manual configuration or help of a server, such as a Dynamic Host Configuration Protocol (DHCP) server. With IPv6, a router on the link advertises in RA messages any global prefixes, and its willingness to function as a default router for the link. RA messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

Simplified Network Renumbering for IPv6 Hosts

The strict aggregation of the global routing table requires that networks be renumbered when the service provider for the network is changed. When the stateless autoconfiguration functionality in IPv6 is used to renumber a network, the prefix from a new service provider is added to RA messages that are sent on the link. (The RA messages contain both the prefix from the old service provider and the prefix from the new service provider.) Nodes on the link automatically configure additional addresses by using the prefix from the new service provider. The nodes can then use the addresses created from the new prefix and the existing addresses created from the old prefix on the link. Configuration of the lifetime parameters associated with the old and new prefixes means that nodes on the link can make the transition to using only addresses created from the new prefix. During a transition period, the old prefix is removed from RA messages and only addresses that contain the new prefix are used on the link (the renumbering is complete) (see [Figure 15](#)).

Figure 15 IPv6 Network Renumbering for Hosts Using Stateless Autoconfiguration



IPv6 General Prefixes

The upper 64 bits of an IPv6 address are composed from a global routing prefix plus a subnet ID, as defined in RFC 3513. A general prefix (for example, /48) holds a short prefix, based on which a number of longer, more specific prefixes (for example, /64) can be defined. When the general prefix is changed, all of the more specific prefixes based on it will change, too. This function greatly simplifies network renumbering and allows for automated prefix definition.

For example, a general prefix might be 48 bits long (“/48”) and the more specific prefixes generated from it might be 64 bits long (“/64”). In the following example, the leftmost 48 bits of all the specific prefixes will be the same—and the same as the general prefix itself. The next 16 bits are all different.

- General prefix: 2001:0DB8:2222::/48
- Specific prefix: 2001:0DB8:2222:0000::/64
- Specific prefix: 2001:0DB8:2222:0001::/64
- Specific prefix: 2001:0DB8:2222:4321::/64
- Specific prefix: 2001:0DB8:2222:7744::/64

General prefixes can be defined in several ways:

- Manually
- Based on a 6to4 interface
- Dynamically, from a prefix received by a DHCP for IPv6 prefix delegation client

More specific prefixes, based on a general prefix, can be used when configuring IPv6 on an interface.

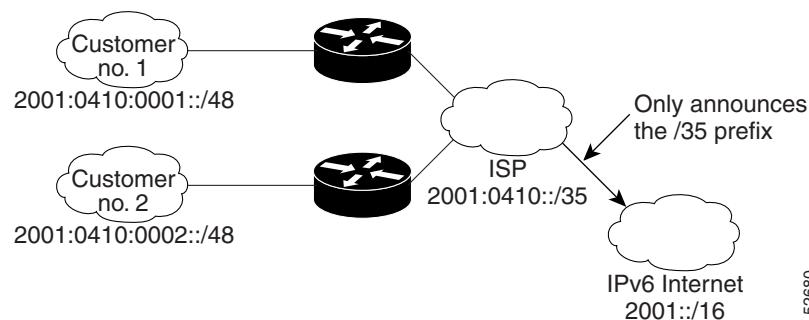
DHCP for IPv6 Prefix Delegation

DHCP for IPv6 can be used in environments to deliver stateful and stateless information. For further information about this feature, see [Implementing DHCP for IPv6](#).

IPv6 Prefix Aggregation

The aggregatable nature of the IPv6 address space enables an IPv6 addressing hierarchy. For example, an enterprise can subdivide a single IPv6 prefix from a service provider into multiple, longer prefixes for use within its internal network. Conversely, a service provider can aggregate all of the prefixes of its customers into a single, shorter prefix that the service provider can then advertise over the IPv6 internet (see [Figure 16](#)).

Figure 16 IPv6 Prefix Aggregation

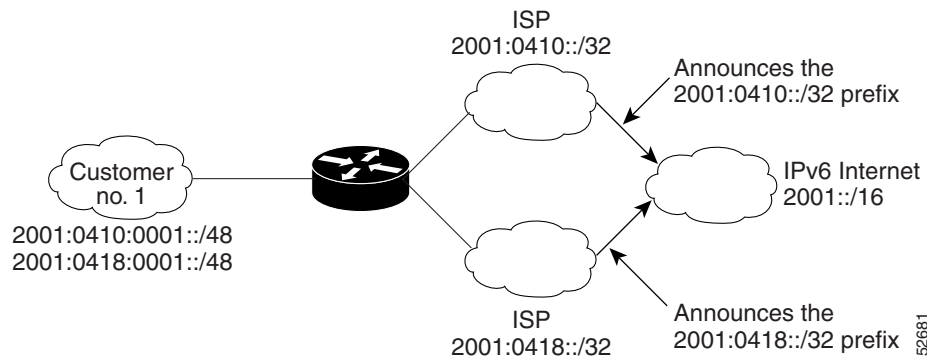


52680

IPv6 Site Multihoming

Multiple IPv6 prefixes can be assigned to networks and hosts. Having multiple prefixes assigned to a network makes it easy for that network to connect to multiple ISPs without breaking the global routing table (see [Figure 17](#)).

Figure 17 IPv6 Site Multihoming



IPv6 Data Links

In IPv6 networks, a data link is a network sharing a particular link-local prefix. Data links are networks arbitrarily segmented by a network administrator in order to provide a multilevel, hierarchical routing structure while shielding the subnetwork from the addressing complexity of attached networks. The function of a subnetwork in IPv6 is similar to a subnetwork in IPv4. A subnetwork prefix is associated with one data link; multiple subnetwork prefixes may be assigned to the same data link.

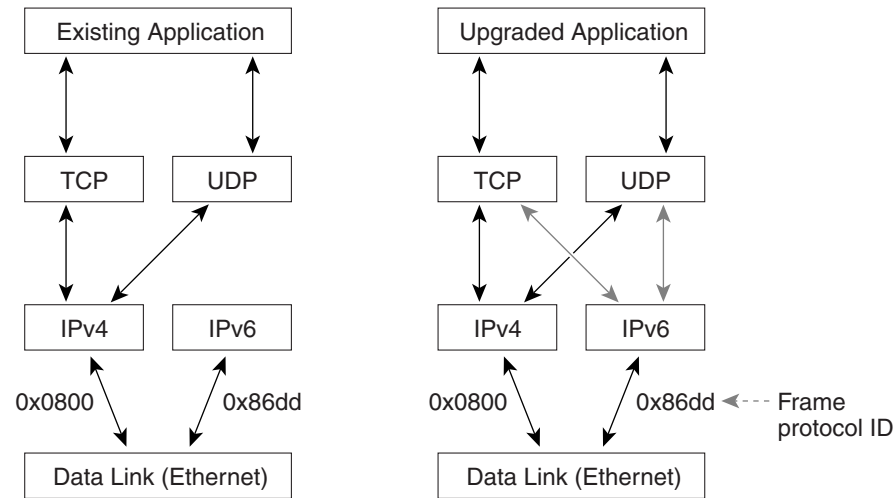
The following data links are supported for IPv6: ATM permanent virtual circuit (PVC) and ATM LANE, Ethernet, Fast Ethernet, Gigabit Ethernet, FDDI, Frame Relay PVC, Cisco High-Level Data Link Control (HDLC), PPP over Packet over SONET, ISDN, serial interfaces, and dynamic packet transport (DPT). See [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#) for release details on supported data links.

Routed Bridge Encapsulation for IPv6

Routed bridge encapsulation (RBE) provides a mechanism for routing a protocol from a bridged interface to another routed or bridged interface. RBE for IPv6 can be used on ATM point-to-point subinterfaces that are configured for IPv6 half-bridging. Routing of IP packets and IPv6 half-bridging, bridging, PPP over Ethernet (PPPoE), or other Ethernet 802.3-encapsulated protocols can be configured on the same subinterface.

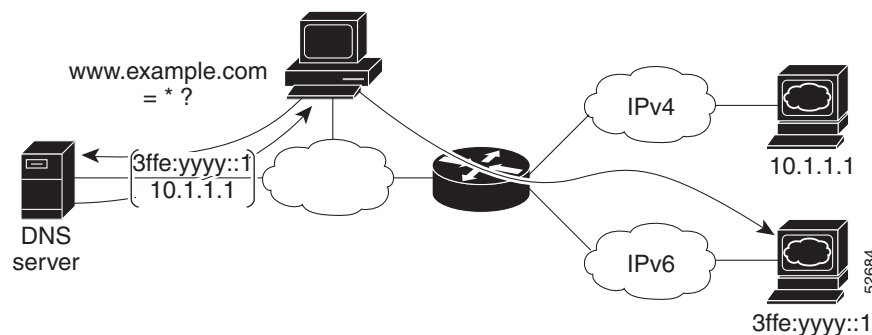
Dual IPv4 and IPv6 Protocol Stacks

The dual IPv4 and IPv6 protocol stack technique can be used to transition to IPv6. It enables gradual, one-by-one upgrades to applications running on nodes. Applications running on nodes are upgraded to make use of the IPv6 protocol stack. Applications that are not upgraded—they support only the IPv4 protocol stack—can coexist with upgraded applications on a node. New and upgraded applications make use of both the IPv4 and IPv6 protocol stacks (see [Figure 18](#)).

Figure 18 *Dual IPv4 and IPv6 Protocol Stack Technique*

One application program interface (API) supports both IPv4 and IPv6 addresses and DNS requests. An application can be upgraded to the new API and still use only the IPv4 protocol stack. The Cisco IOS software supports the dual IPv4 and IPv6 protocol stack technique. When an interface is configured with both an IPv4 and an IPv6 address, the interface will forward both IPv4 and IPv6 traffic.

In [Figure 19](#), an application that supports dual IPv4 and IPv6 protocol stacks requests all available addresses for the destination hostname `www.a.com` from a DNS server. The DNS server replies with all available addresses (both IPv4 and IPv6 addresses) for `www.example.com`. The application chooses an address—in most cases, IPv6 addresses are the default choice—and connects the source node to the destination using the IPv6 protocol stack.

Figure 19 *Dual IPv4 and IPv6 Protocol Stack Applications*

How to Implement IPv6 Addressing and Basic Connectivity

The tasks in the following sections explain how to implement IPv6 addressing and basic connectivity:

- [Configuring IPv6 Addressing and Enabling IPv6 Routing, page 28](#)
- [Defining and Using IPv6 General Prefixes, page 30](#)
- [Configuring an Interface to Support the IPv4 and IPv6 Protocol Stacks, page 33](#)

- [Configuring IPv6 ICMP Rate Limiting, page 34](#)
- [Configuring the DRP Extension for Traffic Engineering, page 35](#)
- [Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36](#)
- [Mapping Hostnames to IPv6 Addresses, page 41](#)
- [Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces, page 42](#)
- [Displaying IPv6 Redirect Messages, page 45](#)

Configuring IPv6 Addressing and Enabling IPv6 Routing

This task explains how to assign IPv6 addresses to individual router interfaces and enable the forwarding of IPv6 traffic globally on the router. By default, IPv6 addresses are not configured and IPv6 routing is disabled.



Note

The *ipv6-address* argument in the **ipv6 address** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

The *ipv6-prefix* argument in the **ipv6 address** command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

The *lprefix-length* keyword and argument in the **ipv6 address** command is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

IPv6 Multicast Groups

An IPv6 address must be configured on an interface for the interface to forward IPv6 traffic. Configuring a global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:1:FF00::/104 for each unicast and anycast address assigned to the interface
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2



Note

The solicited-node multicast address is used in the neighbor discovery process.

Restrictions

In Cisco IOS Release 12.2(4)T or later releases, Cisco IOS Release 12.0(21)ST, and Cisco IOS Release 12.0(22)S or later releases, the **ipv6 address** or **ipv6 address eui-64** command can be used to configure multiple IPv6 global addresses within the same prefix on an interface. Multiple IPv6 link-local addresses on an interface are not supported.

Prior to Cisco IOS Releases 12.2(4)T, 12.0(21)ST, and 12.0(22)S, the Cisco IOS command-line interface (CLI) displays the following error message when multiple IPv6 addresses within the same prefix on an interface are configured:

Prefix <prefix-number> already assigned to <interface-type>

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** *ipv6-prefix/prefix-length* **eui-64**
or
ipv6 address *ipv6-address/prefix-length* **link-local**
or
ipv6 address *ipv6-prefix/prefix-length* **anycast**
or
ipv6 enable
5. **exit**
6. **ipv6 unicast-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 4	ipv6 address <i>ipv6-prefix/prefix-length</i> eui-64 or ipv6 address <i>ipv6-address/prefix-length</i> link-local or ipv6 address <i>ipv6-prefix/prefix-length</i> anycast or ipv6 enable	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. or Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. or Automatically configures an IPv6 link-local address on the interface while also enabling the interface for IPv6 processing. The link-local address can be used only to communicate with nodes on the same link.
	Example: Router(config-if)# ipv6 address 2001:0DB8:0:1::/64 eui-64 or Example: Router(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local or Example: Router(config-if) ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64 anycast or Example: Router(config-if)# ipv6 enable	<ul style="list-style-type: none"> Specifying the ipv6 address eui-64 command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID. Specifying the ipv6 address link-local command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. Specifying the ipv6 address anycast command adds an IPv6 anycast address.
Step 5	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 6	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.

Defining and Using IPv6 General Prefixes

General prefixes can be defined in several ways:

- Manually
- Based on a 6to4 interface
- Dynamically, from a prefix received by a DHCP for IPv6 prefix delegation client

More specific prefixes, based on a general prefix, can be used when configuring IPv6 on an interface.

The following tasks describe how to define and use IPv6 general prefixes:

- [Defining a General Prefix Manually, page 31](#)
- [Defining a General Prefix Based on a 6to4 Interface, page 31](#)
- [Defining a General Prefix with the DHCP for IPv6 Prefix Delegation Client Function, page 32](#)
- [Using a General Prefix in IPv6, page 32](#)

Defining a General Prefix Manually

The following task describes how to define a general prefix manually.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 general-prefix** *prefix-name* [*ipv6-prefix/prefix-length*] [**6to4** *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 general-prefix <i>prefix-name</i> { <i>ipv6-prefix/prefix-length</i> 6to4 <i>interface-type interface-number</i> } Example: Router(config)# ipv6 general-prefix my-prefix 2001:0DB8:2222::/48	Defines a general prefix for an IPv6 address. When defining a general prefix manually, specify both the <i>ipv6-prefix</i> and <i>/prefix-length</i> arguments.

Defining a General Prefix Based on a 6to4 Interface

The following task describes how to define a general prefix based on a 6to4 interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 general-prefix** *prefix-name* [*ipv6-prefix/prefix-length*] [**6to4** *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 general-prefix <i>prefix-name</i> { <i>ipv6-prefix/prefix-length</i> 6to4 <i>interface-type interface-number</i> } Example: Router(config)# ipv6 general-prefix my-prefix 6to4 ethernet 0	Defines a general prefix for an IPv6 address. When defining a general prefix based on a 6to4 interface, specify the 6to4 keyword and the <i>interface-type interface-number</i> arguments. When defining a general prefix based on an interface used for 6to4 tunneling, the general prefix will be of the form 2001:a.b.c.d::/48, where “a.b.c.d” is the IPv4 address of the interface referenced.

Defining a General Prefix with the DHCP for IPv6 Prefix Delegation Client Function

You can define a general prefix dynamically using the DHCP for IPv6 prefix delegation client function. For information on how to perform this task, see the [Implementing DHCP for IPv6](#) module.

Using a General Prefix in IPv6

The following task describes how to use a general prefix in IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** { *ipv6-address/prefix-length* \ *prefix-name sub-bits/prefix-length* }

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if) ipv6 address my-prefix 2001:0DB8:0:7272::/64	Configures an IPv6 prefix name for an IPv6 address and enables IPv6 processing on the interface.

Configuring an Interface to Support the IPv4 and IPv6 Protocol Stacks

When an interface in a Cisco networking device is configured with both an IPv4 and an IPv6 address, the interface forwards both IPv4 and IPv6 traffic—the interface can send and receive data on both IPv4 and IPv6 networks. To configure an interface in a Cisco networking device to support both the IPv4 and IPv6 protocol stacks, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ip address** *ip-address mask* [secondary]
6. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	interface <i>type number</i> Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 5	ip address <i>ip-address mask</i> [secondary [vrf <i>vrf-name</i>]] Example: Router(config-if)# ip address 192.168.99.1 255.255.255.0	Specifies a primary or secondary IPv4 address for an interface.
Step 6	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if)# ipv6 address 2001:0DB8:c18:1::3/64	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface. Note See the “Configuring IPv6 Addressing and Enabling IPv6 Routing” section for more information on configuring IPv6 addresses.

Configuring IPv6 ICMP Rate Limiting

This task explains how to customize IPv6 ICMP rate limiting.

IPv6 ICMP Rate Limiting

In Cisco IOS Release 12.2(8)T or later releases, the IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The initial implementation of IPv6 ICMP rate limiting defined a fixed interval between error messages, but some applications, such as traceroute, often require replies to a group of requests sent in rapid succession. The fixed interval between error messages is not flexible enough to work with applications such as traceroute and can cause the application to fail. Implementing a token bucket scheme allows a number of tokens—representing the ability to send one error message each—to be stored in a virtual bucket. The maximum number of tokens allowed in the bucket can be specified, and for every error message to be sent, one token is removed from the bucket. If a series of error messages is generated,

error messages can be sent until the bucket is empty. When the bucket is empty of tokens, IPv6 ICMP error messages are not sent until a new token is placed in the bucket. The token bucket algorithm does not increase the average rate limiting time interval, and it is more flexible than the fixed time interval scheme.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 icmp error-interval** *milliseconds* [*bucketsize*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 icmp error-interval <i>milliseconds</i> [<i>bucketsize</i>] Example: Router(config)# ipv6 icmp error-interval 50 20	Configures the interval and bucket size for IPv6 ICMP error messages. <ul style="list-style-type: none"> The <i>milliseconds</i> argument specifies the interval between tokens being added to the bucket. The optional <i>bucketsize</i> argument defines the maximum number of tokens stored in the bucket.

Configuring the DRP Extension for Traffic Engineering

This task describes how to configure the DRP extension to RAs in order to signal the preference value of a default router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd router-preference** { **high** | **medium** | **low** }

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 nd router-preference { high medium low } Example: Router(config-if)# ipv6 nd router-preference high	Configures a DRP for a router on a specific interface

Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6

The following tasks explain how to configure Cisco Express Forwarding and distributed Cisco Express Forwarding switching for IPv6:

- [Configuring Cisco Express Forwarding Switching on Distributed and Nondistributed Architecture Platforms, page 36](#)
- [Configuring Unicast RPF, page 39](#)

Configuring Cisco Express Forwarding Switching on Distributed and Nondistributed Architecture Platforms

Cisco Express Forwarding is designed for nondistributed architecture platforms, such as the Cisco 7200 series routers. Distributed Cisco Express Forwarding is designed for distributed architecture platforms, such as the Cisco 12000 series Internet routers or the Cisco 7500 series routers. Nondistributed platforms do not support distributed Cisco Express Forwarding; however, some distributed platforms, such as the Cisco 7500 series routers, support both Cisco Express Forwarding and distributed Cisco Express Forwarding.

When Cisco Express Forwarding is configured on Cisco 7500 series routers, Cisco Express Forwarding switching is performed by the RP; when distributed Cisco Express Forwarding is configured, Cisco Express Forwarding switching is performed by the line cards. By default, the Cisco 12000 series Internet routers support only distributed Cisco Express Forwarding (Cisco Express Forwarding switching is performed by the line cards).

Prerequisites

To enable the router to forward Cisco Express Forwarding and distributed Cisco Express Forwarding traffic, use the **ipv6 unicast-routing** command to configure the forwarding of IPv6 unicast datagrams globally on the router, and use the **ipv6 address** command to configure IPv6 address and IPv6 processing on an interface.

You must enable Cisco Express Forwarding for IPv4 globally on the router by using the **ip cef** command before enabling Cisco Express Forwarding for IPv6 globally on the router.

You must enable distributed Cisco Express Forwarding for IPv4 by using the **ip cef distributed** command before enabling distributed Cisco Express Forwarding for IPv6.

Restrictions

The **ipv6 cef** and **ipv6 cef distributed** commands are not supported on the Cisco 12000 series Internet routers because this distributed platform operates only in distributed Cisco Express Forwarding mode.

In Cisco IOS Release 12.0(22)S or later releases, the following restrictions apply to nondistributed and distributed architecture platforms configured for Cisco Express Forwarding and distributed Cisco Express Forwarding:



Note By default, the Cisco 12000 series Internet routers support only distributed Cisco Express Forwarding (Cisco Express Forwarding switching is performed by the line cards).

- IPv6 packets that have global source and destination addresses are Cisco Express Forwarding-switched or distributed Cisco Express Forwarding-switched.
- IPv6 packets that have link-local source and destination addresses are process-switched.
- IPv6 packets that are tunneled within manually configured IPv6 tunnels are Cisco Express Forwarding-switched.
- Only the following interface and encapsulation types are supported:
 - ATM PVC and ATM LANE
 - Cisco HDLC
 - Ethernet, Fast Ethernet, and Gigabit Ethernet
 - FDDI
 - Frame Relay PVC
 - PPP over Packet over SONET, ISDN, and serial (synchronous and asynchronous) interface types
- The following interface and encapsulation types are not supported:
 - HP 100VG-AnyLAN
 - Switched Multimegabit Data Service (SMDS)
 - Token Ring
 - X.25



Note Contact your local Cisco Systems account representative for specific Cisco Express Forwarding and distributed Cisco Express Forwarding hardware restrictions.

SUMMARY STEPS

1. enable
2. configure terminal
3. ipv6 cef
or
ipv6 cef distributed
4. ipv6 cef accounting [non-recursive | per-prefix | prefix-length]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	<pre>ipv6 cef</pre> <p>or</p> <pre>ipv6 cef distributed</pre> <p>Example: Router(config)# ipv6 cef or</p> <p>Example: Router(config)# ipv6 cef distributed</p>	<p>Enables Cisco Express Forwarding globally on the router.</p> <p>or</p> <p>Enables distributed Cisco Express Forwarding globally on the router.</p>
Step 4	<pre>ipv6 cef accounting [non-recursive per-prefix prefix-length]</pre> <p>Example: Router(config)# ipv6 cef accounting</p>	<p>Enables Cisco Express Forwarding and distributed Cisco Express Forwarding network accounting globally on the router.</p> <ul style="list-style-type: none"> • Network accounting for Cisco Express Forwarding and distributed Cisco Express Forwarding enables you to better understand Cisco Express Forwarding traffic patterns within your network by collecting statistics specific to Cisco Express Forwarding and distributed Cisco Express Forwarding traffic. For example, network accounting for Cisco Express Forwarding and distributed Cisco Express Forwarding enables you to collect information such as the number of packets and bytes switched to a destination or the number of packets switched through a destination. • The optional per-prefix keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 destination (or IPv6 prefix). • The optional prefix-length keyword enables the collection of the number of packets and bytes express forwarded to an IPv6 prefix length. <p>Note When Cisco Express Forwarding is enabled globally on the router, accounting information is collected at the RP; when distributed Cisco Express Forwarding is enabled globally on the router, accounting information is collected at the line cards.</p>

Configuring Unicast RPF

This task explains how to configure unicast RPF.

Prerequisites

To use Unicast RPF, enable Cisco Express Forwarding switching or distributed Cisco Express Forwarding switching in the router. There is no need to configure the input interface for Cisco Express Forwarding switching. As long as Cisco Express Forwarding is running on the router, individual interfaces can be configured with other switching modes.

**Note**

It is very important for Cisco Express Forwarding to be configured globally in the router. Unicast RPF will not work without Cisco Express Forwarding.

Restrictions

Unicast RPF should not be used on interfaces that are internal to the network. Internal interfaces are likely to have routing asymmetry, meaning that there are multiple routes to the source of a packet. Unicast RPF should be applied only where there is natural or configured symmetry.

For example, routers at the edge of the network of an ISP are more likely to have symmetrical reverse paths than routers that are in the core of the ISP network. Routers that are in the core of the ISP network have no guarantee that the best forwarding path out of the router will be the path selected for packets returning to the router. Therefore, we do not recommend that you apply Unicast RPF where there is a chance of asymmetric routing. It is simplest to place Unicast RPF only at the edge of a network or, for an ISP, at the customer edge of the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 verify unicast source reachable-via** {*rx* | *any*} [**allow-default**] [**allow-self-ping**] [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface atm 0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 verify unicast source reachable-via { <i>rx</i> <i>any</i> } [allow-default] [allow-self-ping] [<i>access-list-name</i>] Example: Router(config-if)# ipv6 verify unicast source reachable-via any	Verifies that a source address exists in the FIB table and enables Unicast RPF.

Mapping Hostnames to IPv6 Addresses

This task explains how to map hostnames with IPv6 addresses.

Hostname-to-Address Mappings

A name server is used to track information associated with domain names. A name server can maintain a database of hostname-to-address mappings. Each name can map to one or more IPv4 addresses, IPv6 addresses, or both address types. In order to use this service to map domain names to IPv6 addresses, you must specify a name server and enable the DNS—the global naming scheme of the Internet that uniquely identifies network devices.

The Cisco IOS software maintains a cache of hostname-to-address mappings for use by the **connect**, **telnet**, and **ping** commands, related Telnet support operations, and many other commands that generate command output. This cache speeds the conversion of names to addresses.

Similar to IPv4, IPv6 uses a naming scheme that allows a network device to be identified by its location within a hierarchical name space that provides for domains. Domain names are joined with periods (.) as the delimiting characters. For example, Cisco is a commercial organization that is identified by a com domain name, so its domain name is cisco.com. A specific device in this domain, the FTP server, for example, is identified as ftp.cisco.com.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 host** *name* [*port*] *ipv6-address1* [*ipv6-address2...ipv6-address4*]
4. **ip domain name** [**vrf** *vrf-name*] *name*
or
ip domain list [**vrf** *vrf-name*] *name*
5. **ip name-server** [**vrf** *vrf-name*] *server-address1* [*server-address2...server-address6*]
6. **ip domain-lookup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	<pre>ipv6 host name [port] ipv6-address1 [ipv6-address2...ipv6-address4]</pre> <p>Example:</p> <pre>Router(config)# ipv6 host cisco-sj 2001:0DB8:20:1::12</pre>	<p>Defines a static hostname-to-address mapping in the hostname cache.</p> <ul style="list-style-type: none"> Typically, it is easier to refer to network devices by symbolic names rather than numerical addresses (services such as Telnet can use hostnames or addresses). Hostnames and IPv6 addresses can be associated with one another through static or dynamic means. Manually assigning hostnames to addresses is useful when dynamic mapping is not available.
Step 4	<pre>ip domain name [vrf vrf-name] name</pre> <p>or</p> <pre>ip domain list [vrf vrf-name] name</pre> <p>Example:</p> <pre>Router(config)# ip domain-name cisco.com</pre> <p>or</p> <p>Example:</p> <pre>Router(config)# ip domain list cisco1.com</pre>	<p>(Optional) Defines a default domain name that the Cisco IOS software will use to complete unqualified hostnames.</p> <p>or</p> <p>(Optional) Defines a list of default domain names to complete unqualified hostnames.</p> <ul style="list-style-type: none"> You can specify a default domain name that the Cisco IOS software will use to complete domain name requests. You can specify either a single domain name or a list of domain names. Any hostname that does not contain a complete domain name will have the default domain name you specify appended to it before the name is looked up. <p>Note The ip domain name and ip domain list commands are used to specify default domain names that can be used by both IPv4 and IPv6.</p>
Step 5	<pre>ip name-server [vrf vrf-name] server-address1 [server-address2...server-address6]</pre> <p>Example:</p> <pre>Router(config)# ip name-server 2001:0DB8::250:8bff:fee8:f800 2001:0DB8:0:f004::1</pre>	<p>Specifies one or more hosts that supply name information.</p> <ul style="list-style-type: none"> Specifies one or more hosts (up to six) that can function as a name server to supply name information for DNS. <p>Note The <i>server-address</i> argument can be either an IPv4 or IPv6 address.</p>
Step 6	<pre>ip domain-lookup</pre> <p>Example:</p> <pre>Router(config)# ip domain-lookup</pre>	<p>Enables DNS-based address translation.</p> <ul style="list-style-type: none"> DNS is enabled by default.

Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces

This task explains how to map IPv6 addresses to ATM and Frame Relay PVCs. Specifically, the steps in this section explain how to explicitly map IPv6 addresses to the ATM and Frame Relay PVCs used to reach the addresses.

**Note**

This task shows how to configure both ATM and Frame Relay PVCs. Many of the steps are labeled optional because many networks will require only one type of PVC to be configured. The steps in this section are not applicable to ATM LANE.

IPv6 for Cisco IOS Software Support for Wide-Area Networking Technologies

IPv6 for Cisco IOS software supports wide-area networking technologies such as Cisco HDLC, PPP over Packet over SONET (PoS), ISDN, and serial (synchronous and asynchronous) interface types, ATM PVCs, and Frame Relay PVCs. These technologies function the same in IPv6 as they do in IPv4—IPv6 does not enhance the technologies in any way.

IPv6 Addresses and PVCs

Broadcast and multicast are used in LANs to map protocol (network-layer) addresses to the hardware addresses of remote nodes (hosts and routers). Because using broadcast and multicast to map network-layer addresses to hardware addresses in circuit-based WANs such as ATM and Frame Relay networks is difficult to implement, these networks utilize implicit, explicit, and dynamic mappings for the network-layer addresses of remote nodes and the PVCs used to reach the addresses.

Assigning an IPv6 address to an interface by using the **ipv6 address** command defines the IPv6 addresses for the interface and the network that is directly connected to the interface. If only one PVC is terminated on the interface (the interface is a point-to-point interface), there is an implicit mapping between all of the IPv6 addresses on the network and the PVC used to reach the addresses (no additional address mappings are needed). If several PVCs are terminated on the interface (the interface is a point-to-multipoint interface), the **protocol ipv6** command (for ATM networks) or the **frame-relay map ipv6** command (for Frame Relay networks) is used to configure explicit mappings between the IPv6 addresses of the remote nodes and the PVCs used to reach the addresses.

**Note**

Given that IPv6 supports multiple address types, and depending on which applications or protocols are configured on a point-to-multipoint interface, you may need to configure multiple explicit mappings between the IPv6 addresses of the interface and the PVC used to reach the addresses. For example, explicitly mapping both the link-local and global IPv6 address of a point-to-multipoint interface to the PVC that the interface terminates ensures that the Interior Gateway Protocol (IGP) configured on the interface forwards traffic to and from the PVC correctly.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **pvc** [*name*] *vpilvci* [*ces* | *ilmi* | *qsaal* | *smds* | *l2transport*]
5. **protocol ipv6** *ipv6-address* [[**no**] **broadcast**]
6. **exit**
7. **ipv6 address** *ipv6-address/prefix-length* **link-local**
8. **exit**
9. **interface** *type number*

10. **frame-relay map ipv6** *ipv6-address dlc* [**broadcast**] [**cisco**] [**ietf**] [**payload-compression** {**packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}]
11. **ipv6 address** *ipv6-address/prefix-length* **link-local**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface atm 0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	pvc [<i>name</i>] <i>vpi/vci</i> [ces ilmi qsaal smds l2transport] Example: Router(config-if)# pvc 1/32	(Optional) Creates or assigns a name to an ATM PVC and places the router in ATM VC configuration mode.
Step 5	protocol ipv6 <i>ipv6-address</i> [[no] broadcast] Example: Router(config-if-atm-vc)# protocol ipv6 2001:0DB8:2222:1003::45	(Optional) Maps the IPv6 address of a remote node to the PVC used to reach the address. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The optional [no] broadcast keywords indicate whether the map entry should be used when IPv6 multicast packets (not broadcast packets) are sent to the interface. Pseudobroadcasting is supported. The [no] broadcast keywords in the protocol ipv6 command take precedence over the broadcast command configured on the same ATM PVC.
Step 6	exit Example: Router(config-if-atm-vc)# exit	Exits ATM VC configuration mode, and returns the router to interface configuration mode.

	Command or Action	Purpose
Step 7	ipv6 address <i>ipv6-address/prefix-length</i> link-local Example: Router(config-if)# ipv6 address 2001:0DB8:2222:1003::72/64 link-local	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. <ul style="list-style-type: none"> In the context of this task, a link-local address of the node at the other end of the link is required for the IGP used in the network. Specifying the ipv6 address link-local command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.
Step 8	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 9	interface <i>type number</i> Example: Router(config)# interface serial 3	Specifies an interface type and number, and places the router in interface configuration mode.
Step 10	frame-relay map ipv6 <i>ipv6-address dlci</i> [broadcast] [cisco] [ietf] [payload-compression] { packet-by-packet frf9 stac [<i>hardware-options</i>] data-stream stac [<i>hardware-options</i>]}]	(Optional) Maps the IPv6 address of a remote node to the data-link connection identifier (DLCI) of the PVC used to reach the address.
Step 11	ipv6 address <i>ipv6-address/prefix-length</i> link-local Example: Router(config-if)# ipv6 address 2001:0DB8:2222:1044::46/64 link-local	Specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface. <ul style="list-style-type: none"> In the context of this task, a link-local address of the node at the other end of the link is required for the IGP used in the network. Specifying the ipv6 address link-local command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface.

Displaying IPv6 Redirect Messages

This task explains how to display IPv6 redirect messages. The commands shown are optional and can be entered in any order.

IPv6 Redirect Messages

The IPv6 Redirect Messages feature enables a router to send ICMP IPv6 neighbor redirect messages to inform hosts of better first hop nodes (routers or hosts) on the path to a destination.

There are no configuration tasks for the IPv6 Redirect Messages feature. The sending of IPv6 redirect messages is enabled by default. Use the **no ipv6 redirects** command to disable the sending of IPv6 redirect messages on an interface. Use the **ipv6 redirects** command to reenale the sending of IPv6 redirect messages if the Cisco IOS software is forced to resend a packet through the same interface on which the packet was received.

To verify whether the sending of IPv6 redirect messages is enabled on an interface, enter the **show ipv6 interface** command.

SUMMARY STEPS

1. **enable**
2. **show ipv6 interface** [**brief**] [*type number*] [**prefix**]
3. **show ipv6 neighbors** [*interface-type interface-number* | *ipv6-address* | *ipv6-hostname* | **statistics**]
4. **show ipv6 route** [*ipv6-address* | *ipv6-prefix/prefix-length* | *protocol* | *interface-type interface-number*]
5. **show ipv6 traffic**
6. **show frame-relay map** [**interface** *type number*] [*dlci*]
7. **show atm map**
8. **show hosts** [**vrf** *vrf-name* | **all** | *hostname* | **summary**]
9. **enable**
10. **show running-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router# enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ipv6 interface [brief] [<i>type number</i>] [prefix] Example: Router# show ipv6 interface ethernet 0	Displays the usability status of interfaces configured for IPv6. <ul style="list-style-type: none"> • Displays information about the status of IPv6 neighbor redirect messages, IPv6 neighbor discovery messages, and stateless autoconfiguration.
Step 3	show ipv6 neighbors [<i>interface-type interface-number</i> <i>ipv6-address</i> <i>ipv6-hostname</i> statistics] Example: Router# show ipv6 neighbors ethernet 2	Displays IPv6 neighbor discovery cache information.

	Command or Action	Purpose
Step 4	show ipv6 route [<i>ipv6-address</i> <i>ipv6-prefix/prefix-length</i> <i>protocol</i> <i>interface-type interface-number</i>] Example: Router# show ipv6 route	(Optional) Displays the current contents of the IPv6 routing table.
Step 5	show ipv6 traffic Example: Router# show ipv6 traffic	(Optional) Displays statistics about IPv6 traffic.
Step 6	show frame-relay map [<i>interface type number</i>] [<i>dlci</i>] Example: Router# show frame-relay map	Displays the current map entries and information about the Frame Relay connections.
Step 7	show atm map Example: Router# show atm map	Displays the list of all configured ATM static maps to remote hosts on an ATM network and on ATM bundle maps.
Step 8	show hosts [<i>vrf vrf-name</i> all <i>hostname</i> summary] Example: Router# show hosts	Displays the default domain name, the style of name lookup service, a list of name server hosts, and the cached list of hostnames and addresses.
Step 9	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 10	show running-config Example: Router# show running-config	Displays the current configuration running on the router.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 interface Command](#)
- [Sample Output from the show ipv6 neighbors Command](#)
- [Sample Output from the show ipv6 route Command](#)
- [Sample Output from the show ipv6 traffic Command](#)
- [Sample Output from the show frame-relay map Command](#)
- [Sample Output from the show atm map Command](#)
- [Sample Output from the show hosts Command](#)
- [Sample Output from the show running-config Command](#)

Sample Output from the show ipv6 interface Command

In the following example, the **show ipv6 interface** command is used to verify that IPv6 addresses are configured correctly for Ethernet interface 0. Information is also displayed about the status of IPv6 neighbor redirect messages, IPv6 neighbor discovery messages, and stateless autoconfiguration.

```
Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
IPv6 is stalled, link-local address is FE80::1
Global unicast address(es):
  2001:0DB8:2000::1, subnet is 2001:0DB8:2000::/64
  2001:0DB8:3000::1, subnet is 2001:0DB8:3000::/64
Joined group address(es):
  FF02::1
  FF02::2
  FF02::1:FF00:1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

Sample Output from the show ipv6 neighbors Command

In the following example, the **show ipv6 neighbors** command is used to display IPv6 neighbor discovery cache information. A hyphen (-) in the Age field of the command output indicates a static entry. The following example displays IPv6 neighbor discovery cache information for Ethernet interface 2:

```
Router# show ipv6 neighbors ethernet 2

IPv6 Address                               Age Link-layer Addr State Interface
2001:0DB8:0:4::2                          0 0003.a0d6.141e REACH Ethernet2
FE80::XXXX:A0FF:FED6:141E                  0 0003.a0d6.141e REACH Ethernet2
2001:0DB8:1::45a                          - 0002.7d1a.9472 REACH Ethernet2
```

Sample Output from the show ipv6 route Command

When the *ipv6-address* or *ipv6-prefix/prefix-length* argument is specified, only route information for that address or network is displayed. The following is sample output from the **show ipv6 route** command when entered with the IPv6 prefix 2001:0DB8::/35:

```
Router# show ipv6 route 2001:0DB8::/35

IPv6 Routing Table - 261 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea

B 2001:0DB8::/35 [20/3]
  via FE80::60:5C59:9E00:16, Tunnel1
```

Sample Output from the show ipv6 traffic Command

In the following example, the **show ipv6 traffic** command is used to display ICMP rate-limited counters:

```
Router# show ipv6 traffic

ICMP statistics:
  Rcvd: 188 input, 0 checksum errors, 0 too short
        0 unknown info type, 0 unknown error type
```

```

unreach: 0 routing, 0 admin, 0 neighbor, 0 address, 0 port
parameter: 0 error, 0 header, 0 option
0 hopcount expired, 0 reassembly timeout, 0 too big
0 echo request, 0 echo reply
0 group query, 0 group report, 0 group reduce
1 router solicit, 175 router advert, 0 redirects
0 neighbor solicit, 12 neighbor advert
Sent: 7376 output, 56 rate-limited
unreach: 0 routing, 15 admin, 0 neighbor, 0 address, 0 port
parameter: 0 error, 0 header, 0 option
0 hopcount expired, 0 reassembly timeout, 0 too big
15 echo request, 0 echo reply
0 group query, 0 group report, 0 group reduce
0 router solicit, 7326 router advert, 0 redirects
2 neighbor solicit, 22 neighbor advert

```

Sample Output from the show frame-relay map Command

In the following example, the **show frame-relay map** command is used to verify that the IPv6 address of a remote node is mapped to the DLCI of the PVC used to reach the address. The following example shows that the link-local and global IPv6 addresses (FE80::E0:F727:E400:A and 2001:0DB8:2222:1044::73; FE80::60:3E47:AC8:8 and 2001:0DB8:2222:1044::72) of two remote nodes are explicitly mapped to DLCI 17 and DLCI 19, respectively. Both DLCI 17 and DLCI 19 are terminated on interface serial 3 of this node; therefore, interface serial 3 of this node is a point-to-multipoint interface.

Router# **show frame-relay map**

```

Serial3 (up): ipv6 FE80::E0:F727:E400:A dlci 17(0x11,0x410), static,
              broadcast, CISCO, status defined, active
Serial3 (up): ipv6 2001:0DB8:2222:1044::72 dlci 19(0x13,0x430), static,
              CISCO, status defined, active
Serial3 (up): ipv6 2001:0DB8:2222:1044::73 dlci 17(0x11,0x410), static,
              CISCO, status defined, active
Serial3 (up): ipv6 FE80::60:3E47:AC8:8 dlci 19(0x13,0x430), static,
              broadcast, CISCO, status defined, active

```

Sample Output from the show atm map Command

In the following example, the **show atm map** command is used to verify that the IPv6 address of a remote node is mapped to the PVC used to reach the address. The following example shows that the link-local and global IPv6 addresses (FE80::60:3E47:AC8:C and 2001:0DB8:2222:1003::72, respectively) of a remote node are explicitly mapped to PVC 1/32 of ATM interface 0:

Router# **show atm map**

```

Map list ATM0pvc1 : PERMANENT
ipv6 FE80::60:3E47:AC8:C maps to VC 1, VPI 1, VCI 32, ATM0
, broadcast
ipv6 2001:0DB8:2222:1003::72 maps to VC 1, VPI 1, VCI 32, ATM0

```

Sample Output from the show hosts Command

The state of the name lookup system on the DHCP for IPv6 client can be displayed with the **show hosts** command:

Router# **show hosts**

```

Default domain is not set
Domain list:example.com
Name/address lookup uses domain service
Name servers are 2001:0DB8:A:B::1, 2001:0DB8:3000:3000::42

```

```
Codes:UN - unknown, EX - expired, OK - OK, ?? - revalidate
      temp - temporary, perm - permanent
      NA - Not Applicable None - Not defined
```

Host	Port	Flags	Age	Type	Address(es)
sdfasfd	None	(temp, UN)	0	IPv6	

Sample Output from the show running-config Command

In the following example, the **show running-config** command is used to verify that IPv6 processing of packets is enabled globally on the router and on applicable interfaces, and that an IPv6 address is configured on applicable interfaces:

```
Router# show running-config
```

```
Building configuration...
```

```
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ipv6 unicast-routing
!
interface Ethernet0
  no ip route-cache
  no ip mroute-cache
  no keepalive
  media-type 10BaseT
  ipv6 address 2001:0DB8:0:1::/64 eui-64
!
```

In the following example, the **show running-config** command is used to verify that Cisco Express Forwarding and network accounting for Cisco Express Forwarding have been enabled globally on a nondistributed architecture platform, and that Cisco Express Forwarding has been enabled on an IPv6 interface. The following output shows that both that Cisco Express Forwarding and network accounting for Cisco Express Forwarding have been enabled globally on the router, and that Cisco Express Forwarding has also been enabled on Ethernet interface 0:

```
Router# show running-config
```

```
Building configuration...
```

```
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface Ethernet0
  ip address 10.4.9.11 255.0.0.0
  media-type 10BaseT
  ipv6 address 2001:0DB8:C18:1::/64 eui-64
!
```


In the following example, the **show running-config** command is used to verify that distributed Cisco Express Forwarding and network accounting for distributed Cisco Express Forwarding have been enabled globally on a distributed architecture platform, such as the Cisco 7500 series routers. The following example shows that both distributed Cisco Express Forwarding and network accounting for Cisco Express Forwarding have been enabled globally on the router.

**Note**

Distributed Cisco Express Forwarding is enabled by default on the Cisco 12000 series Internet routers and disabled by default on the Cisco 7500 series routers. Therefore, output from the **show running-config** command on the Cisco 12000 series does not show whether distributed Cisco Express Forwarding is configured globally on the router. The following output is from a Cisco 7500 series router.

```
Router# show running-config

Building configuration...

Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ip cef distributed
ipv6 unicast-routing
ipv6 cef distributed
ipv6 cef accounting prefix-length
```

In the following example, the **show running-config** command is used to verify static hostname-to-address mappings, default domain names, and name servers in the hostname cache, and to verify that the DNS service is enabled:

```
Router# show running-config

Building configuration...
!
ipv6 host cisco-sj 2001:0DB8:20:1::12
!
ip domain-name cisco.com
ip domain-lookup
ip name-server 2001:0DB8:C01F:768::1
```

Configuration Examples for Implementing IPv6 Addressing and Basic Connectivity

This section provides the following configuration examples:

- [IPv6 Addressing and IPv6 Routing Configuration: Example, page 52](#)
- [Dual Protocol Stacks Configuration: Example, page 52](#)
- [IPv6 ICMP Rate Limiting Configuration: Example, page 52](#)
- [Cisco Express Forwarding and Distributed Cisco Express Forwarding Configuration: Example, page 53](#)
- [Hostname-to-Address Mappings Configuration: Example, page 53](#)
- [IPv6 Address to ATM and Frame Relay PVC Mapping Configuration: Examples, page 53](#)

IPv6 Addressing and IPv6 Routing Configuration: Example

In the following example, IPv6 is enabled on the router with both a link-local address and a global address based on the IPv6 prefix 2001:0DB8:c18:1::/64. The EUI-64 interface ID is used in the low-order 64 bits of both addresses. Output from the **show ipv6 interface** command is included to show how the interface ID (260:3EFF:FE47:1530) is appended to the link-local prefix FE80::/64 of Ethernet interface 0.

```
ipv6 unicast-routing

interface ethernet 0
  ipv6 address 2001:0DB8:c18:1::/64 eui-64

Router# show ipv6 interface ethernet 0

Ethernet0 is up, line protocol is up
IPv6 is enabled, link-local address is FE80::260:3EFF:FE47:1530
Global unicast address(es):
  2001:0DB8:C18:1:260:3EFF:FE47:1530, subnet is 2001:0DB8:C18:1::/64
Joined group address(es):
  FE02::1
  FE02::2
  FE02::1:FF47:1530
  FE02::9
MTU is 1500 bytes
ICMP error messages limited to one every 500 milliseconds
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

In the following example, multiple IPv6 global addresses within the prefix 2001:0DB8::/64 are configured on Ethernet interface 0:

```
interface ethernet 0
  ipv6 address 2001:0DB8::1/64
  ipv6 address 2001:0DB8::/64 eui-64
```

Dual Protocol Stacks Configuration: Example

The following example enables the forwarding of IPv6 unicast datagrams globally on the router and configures Ethernet interface 0 with both an IPv4 address and an IPv6 address:

```
ipv6 unicast-routing

interface Ethernet0
  ip address 192.168.99.1 255.255.255.0
  ipv6 address 2001:0DB8:c18:1::3/64
```

IPv6 ICMP Rate Limiting Configuration: Example

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
ipv6 icmp error-interval 50 20
```

Cisco Express Forwarding and Distributed Cisco Express Forwarding Configuration: Example

In the following example, both Cisco Express Forwarding for IPv6 and network accounting for Cisco Express Forwarding for IPv6 have been enabled globally on a nondistributed architecture router, and Cisco Express Forwarding for IPv6 has been enabled on Ethernet interface 0. The example also shows that the forwarding of IPv6 unicast datagrams has been configured globally on the router with the **ipv6 unicast-routing** command, an IPv6 address has been configured on Ethernet interface 0 with the **ipv6 address** command, and Cisco Express Forwarding for IPv4 has been configured globally on the router with the **ip cef** command.

```
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length

interface Ethernet0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:0DB8:C18:1::/64 eui-64
```

In the following example, both distributed Cisco Express Forwarding for IPv6 and network accounting for distributed Cisco Express Forwarding for IPv6 have been enabled globally on a distributed architecture router. The forwarding of IPv6 unicast datagrams has been configured globally on the router with the **ipv6 unicast-routing** command and distributed Cisco Express Forwarding for IPv4 has been configured globally on the router with the **ip cef distributed** command.

```
ip cef distributed
ipv6 unicast-routing
ipv6 cef distributed
ipv6 cef accounting prefix-length
```

Hostname-to-Address Mappings Configuration: Example

The following example defines two static hostname-to-address mappings in the hostname cache, establishes a domain list with several alternate domain names to complete unqualified hostnames, specifies host 2001:0DB8::250:8bff:fee8:f800 and host 2001:0DB8:0:f004::1 as the name servers, and reenables the DNS service:

```
ipv6 host cisco-sj 2001:0DB8:700:20:1::12
ipv6 host cisco-hq 2001:0DB8:768::1 2001:0DB8:20:1::22
ip domain list example1.com
ip domain list example2.com
ip domain list example3.edu
ip name-server 2001:0DB8::250:8bff:fee8:f800 2001:0DB8:0:f004::1
ip domain-lookup
```

IPv6 Address to ATM and Frame Relay PVC Mapping Configuration: Examples

This section provides the following IPv6 ATM and Frame Relay PVC mapping configuration examples:

- [IPv6 ATM PVC Mapping Configuration—Point-to-Point Interface: Example](#)
- [IPv6 ATM PVC Mapping Configuration—Point-to-Multipoint Interface: Example](#)
- [IPv6 Frame Relay PVC Mapping Configuration—Point-to-Point Interface: Example](#)

- [IPv6 Frame Relay PVC Mapping Configuration—Point-to-Multipoint Interface: Example](#)

IPv6 ATM PVC Mapping Configuration—Point-to-Point Interface: Example

In the following example, two nodes named Router 1 and Router 2 are connected by a single PVC. The point-to-point subinterface ATM0.132 is used on both nodes to terminate the PVC; therefore, the mapping between the IPv6 addresses of both nodes and the PVC is implicit (no additional mappings are required).

Router 1 Configuration

```
interface ATM 0
  no ip address
!
interface ATM 0.132 point-to-point
  pvc 1/32
  encapsulation aal5snap
!
  ipv6 address 2001:0DB8:2222:1003::72/64
```

Router 2 Configuration

```
interface ATM 0
  no ip address
!
interface ATM 0.132 point-to-point
  pvc 1/32
  encapsulation aal5snap
!
  ipv6 address 2001:0DB8:2222:1003::45/64
```

IPv6 ATM PVC Mapping Configuration—Point-to-Multipoint Interface: Example

In the following example, the same two nodes (Router 1 and Router 2) from the previous example are connected by the same PVC. In this example, however, the point-to-multipoint interface ATM0 is used on both nodes to terminate the PVC; therefore, explicit mappings are required between the link-local and global IPv6 addresses of interface ATM0 on both nodes and the PVC. Additionally, ATM pseudobroadcasts are enabled on the link-local address of interface ATM0 on both nodes. The link-local address specified here is the link-local address of the other end of the PVC.

Router 1 Configuration

```
interface ATM 0
  no ip address
  pvc 1/32
  protocol ipv6 2001:0DB8:2222:1003::45
  protocol ipv6 FE80::60:2FA4:8291:2 broadcast
  encapsulation aal5snap
!
  ipv6 address 2001:0DB8:2222:1003::72/64
```

Router 2 Configuration

```
interface ATM 0
  no ip address
  pvc 1/32
  protocol ipv6 FE80::60:3E47:AC8:C broadcast
  protocol ipv6 2001:0DB8:2222:1003::72
  encapsulation aal5snap
!
```

```
ipv6 address 2001:0DB8:2222:1003::45/64
```

IPv6 Frame Relay PVC Mapping Configuration—Point-to-Point Interface: Example

In the following example, three nodes named Router A, Router B, and Router C make up a fully meshed network. Each node is configured with two PVCs, which provide an individual connection to each of the other two nodes. Each PVC is configured on a different point-to-point subinterface, which creates three unique IPv6 networks (2001:0DB8:2222:1017::/64, 2001:0DB8:2222:1018::/64, and 2001:0DB8:2222:1019::/64). Therefore, the mappings between the IPv6 addresses of each node and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses are implicit (no additional mappings are required).



Note

Given that each PVC in the following example is configured on a different point-to-point subinterface, the configuration in the following example can also be used in a network that is not fully meshed. Additionally, configuring each PVC on a different point-to-point subinterface can help simplify your routing protocol configuration. However, the configuration in the following example requires more than one IPv6 network, whereas configuring each PVC on point-to-multipoint interfaces requires only one IPv6 network.

Router A Configuration

```
interface Serial 3
  encapsulation frame-relay
  !
interface Serial3.17 point-to-point
  description to Router B
  ipv6 address 2001:0DB8:2222:1017::46/64
  frame-relay interface-dlci 17
  !
interface Serial 3.19 point-to-point
  description to Router C
  ipv6 address 2001:0DB8:2222:1019::46/64
  frame-relay interface-dlci 19
```

Router B Configuration

```
interface Serial 5
  encapsulation frame-relay
  !
interface Serial5.17 point-to-point
  description to Router A
  ipv6 address 2001:0DB8:2222:1017::73/64
  frame-relay interface-dlci 17
  !
interface Serial5.18 point-to-point
  description to Router C
  ipv6 address 2001:0DB8:2222:1018::73/64
  frame-relay interface-dlci 18
```

Router C Configuration

```
interface Serial 0
  encapsulation frame-relay
  !
interface Serial0.18 point-to-point
  description to Router B
  ipv6 address 2001:0DB8:2222:1018::72/64
  frame-relay interface-dlci 18
  !
```

```
interface Serial0.19 point-to-point
description to Router A
ipv6 address 2001:0DB8:2222:1019::72/64
frame-relay interface-dlci 19
```

IPv6 Frame Relay PVC Mapping Configuration—Point-to-Multipoint Interface: Example

In the following example, the same three nodes (Router A, Router B, and Router C) from the previous example make up a fully meshed network and each node is configured with two PVCs (which provide an individual connection to each of the other two nodes). However, the two PVCs on each node in the following example are configured on a single interface (serial 3, serial 5, and serial 10, respectively), which makes each interface a point-to-multipoint interface. Therefore, explicit mappings are required between the link-local and global IPv6 addresses of each interface on all three nodes and the DLCI (DLCI 17, 18, and 19) of the PVC used to reach the addresses.

Router A Configuration

```
interface Serial 3
encapsulation frame-relay
ipv6 address 2001:0DB8:2222:1044::46/64
frame-relay map ipv6 FE80::E0:F727:E400:A 17 broadcast
frame-relay map ipv6 FE80::60:3E47:AC8:8 19 broadcast
frame-relay map ipv6 2001:0DB8:2222:1044::72 19
frame-relay map ipv6 2001:0DB8:2222:1044::73 17
```

Router B Configuration

```
interface Serial 5
encapsulation frame-relay
ipv6 address 2001:0DB8:2222:1044::73/64
frame-relay map ipv6 FE80::60:3E59:DA78:C 17 broadcast
frame-relay map ipv6 FE80::60:3E47:AC8:8 18 broadcast
frame-relay map ipv6 2001:0DB8:2222:1044::46 17
frame-relay map ipv6 2001:0DB8:2222:1044::72 18
```

Router C Configuration

```
interface Serial 10
encapsulation frame-relay
ipv6 address 2001:0DB8:2222:1044::72/64
frame-relay map ipv6 FE80::60:3E59:DA78:C 19 broadcast
frame-relay map ipv6 FE80::E0:F727:E400:A 18 broadcast
frame-relay map ipv6 2001:0DB8:2222:1044::46 19
frame-relay map ipv6 2001:0DB8:2222:1044::73 18
```

Where to Go Next

If you want to implement IPv6 routing protocols, see the [Implementing RIP for IPv6](#), [Implementing IS-IS for IPv6](#), or [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing IPv6 Addressing and Basic Connectivity feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IPv6 DHCP description and configuration	“Implementing DHCP for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv4 addressing configuration tasks	“Configuring IPv4 Addresses,” Cisco IOS IP Addressing Services Configuration Guide
IPv4 services configuration tasks	“Configuring IP Services,” Cisco IOS IP Application Services Configuration Guide
IPv4 addressing commands	Cisco IOS IP Addressing Services Command Reference
IPv4 IP services commands	Cisco IOS IP Application Services Command Reference
Stateful switchover	“Stateful Switchover,” Cisco IOS High Availability Configuration Guide
Switching configuration tasks	“Cisco IOS IP Switching Features Roadmap,” Cisco IOS IP Switching Configuration Guide
Switching commands	Cisco IOS IP Switching Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1981	<i>Path MTU Discovery for IP version 6</i>
RFC 2373	<i>IP Version 6 Addressing Architecture</i>
RFC 2374	<i>An Aggregatable Global Unicast Address Format</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2461	<i>Neighbor Discovery for IP Version 6 (IPv6)</i>
RFC 2462	<i>IPv6 Stateless Address Autoconfiguration</i>
RFC 2463	<i>Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification</i>
RFC 2464	<i>Transmission of IPv6 Packets over Ethernet Networks</i>
RFC 2467	<i>Transmission of IPv6 Packets over FDDI Networks</i>
RFC 2472	<i>IP Version 6 over PPP</i>
RFC 2492	<i>IPv6 over ATM Networks</i>
RFC 2590	<i>Transmission of IPv6 Packets over Frame Relay Networks Specification</i>
RFC 3152	<i>Delegation of IP6.ARPA</i>
RFC 3162	<i>RADIUS and IPv6</i>
RFC 3513	<i>Internet Protocol Version 6 (IPv6) Addressing Architecture</i>
RFC 3596	<i>DNS Extensions to Support IP version 6</i>
RFC 3879	<i>Deprecating Site Local Addresses</i>
RFC 4193	<i>Unique Local IPv6 Unicast Addresses</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **atm route-bridged**
- **cef table consistency-check**
- **clear cef table**
- **clear ipv6 neighbors**
- **clear ipv6 route**
- **clear ipv6 traffic**
- **copy**
- **debug adjacency**
- **debug ipv6 cef drop**
- **debug ipv6 cef events**
- **debug ipv6 cef hash**
- **debug ipv6 cef receive**
- **debug ipv6 cef table**
- **debug ipv6 icmp**
- **debug ipv6 nd**
- **debug ipv6 packet**
- **debug ipv6 routing**
- **frame-relay map ipv6**
- **ip name-server**
- **ipv6 address**
- **ipv6 address anycast**
- **ipv6 address eui-64**
- **ipv6 address link-local**
- **ipv6 atm-vc**
- **ipv6 cef**
- **ipv6 cef accounting**
- **ipv6 cef distributed**
- **ipv6 enable**
- **ipv6 general-prefix**
- **ipv6 hop-limit**
- **ipv6 icmp error-interval**

- **ipv6 mtu**
- **ipv6 nd dad attempts**
- **ipv6 nd managed-config-flag**
- **ipv6 nd ns-interval**
- **ipv6 nd prefix**
- **ipv6 nd prefix-advertisement**
- **ipv6 nd ra interval**
- **ipv6 nd ra lifetime**
- **ipv6 nd ra suppress**
- **ipv6 nd reachable-time**
- **ipv6 nd router-preference**
- **ipv6 neighbor**
- **ipv6 redirects**
- **ipv6 unicast-routing**
- **ipv6 unnumbered**
- **ipv6 verify unicast reverse-path**
- **ipv6 verify unicast source reachable-via**
- **logging host**
- **logging origin-id**
- **logging source-interface**
- **neighbor activate**
- **neighbor override-capability-neg**
- **neighbor send-label**
- **neighbor translate-update**
- **neighbor update-source**
- **ping**
- **ping ipv6**
- **protocol ipv6 (ATM)**
- **show adjacency**
- **show atm map**
- **show cdp entry**
- **show cdp neighbors**
- **show cef**
- **show cef interface**
- **show cef linecard**
- **show frame-relay map**
- **show ipv6 cef**
- **show ipv6 cef adjacency**

- **show ipv6 cef non-recursive**
- **show ipv6 cef summary**
- **show ipv6 cef switching statistics**
- **show ipv6 cef traffic prefix-length**
- **show ipv6 cef tree**
- **show ipv6 cef unresolved**
- **show ipv6 general-prefix**
- **show ipv6 interface**
- **show ipv6 mtu**
- **show ipv6 neighbors**

Feature Information for Implementing IPv6 Addressing and Basic Connectivity

Table 5 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

Table 5 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 5 **Feature Information for Implementing IPv6 Addressing and Basic Connectivity**

Feature Name	Releases	Feature Information
IPv6: ICMPv6	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	ICMP for IPv6 generates error messages, such as ICMP destination unreachable messages, and informational messages, such as ICMP echo request and reply messages. Additionally, ICMP packets in IPv6 are used in the IPv6 neighbor discovery process, path MTU discovery, and the MLD protocol for IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> • ICMP for IPv6, page 18 • IPv6 Neighbor Discovery, page 18 • IPv6 Neighbor Solicitation Message, page 19 • IPv6 Router Advertisement Message, page 21 • IPv6 Stateless Autoconfiguration, page 24 • Configuring IPv6 ICMP Rate Limiting, page 34 • IPv6 ICMP Rate Limiting Configuration: Example, page 52
IPv6: ICMPv6 redirect	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(4)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message. Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Neighbor Redirect Message, page 22 • IPv6 Redirect Messages, page 46
IPv6: ICMP rate limiting	12.2(8)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	The IPv6 ICMP rate limiting feature implements a token bucket algorithm for limiting the rate at which IPv6 ICMP error messages are sent out on the network. The following sections provide information about this feature: <ul style="list-style-type: none"> • Configuring IPv6 ICMP Rate Limiting, page 34 • IPv6 ICMP Rate Limiting, page 34 • IPv6 ICMP Rate Limiting Configuration: Example, page 52

Table 5 *Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)*

Feature Name	Releases	Feature Information
IPv6: IPv6 default router preferences	12.2(33)SB 12.2(33)SRA 12.4(2)T 12.2(33)SXH	<p>The DRP extension provides a coarse preference metric (low, medium, or high) for default routers.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Default Router Preferences for Traffic Engineering, page 22 • Configuring the DRP Extension for Traffic Engineering, page 35
IPv6: IPv6 MTU path discovery	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>Path MTU discovery in IPv6 allows a host to dynamically discover and adjust to differences in the MTU size of every link along a given data path.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Path MTU Discovery for IPv6, page 17 • ICMP for IPv6, page 18
IPv6: IPv6 neighbor discovery	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The IPv6 neighbor discovery process uses ICMP messages and solicited-node multicast addresses to determine the link-layer address of a neighbor on the same network (local link), verify the reachability of a neighbor, and track neighboring routers.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Link-Local Address, page 7 • ICMP for IPv6, page 18 • IPv6 Neighbor Discovery, page 18 • IPv6 Multicast Groups, page 28
IPv6: IPv6 neighbor discovery duplicate address detection	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(4)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>IPv6 neighbor discovery duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed).</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Neighbor Solicitation Message, page 19 • IPv6 Stateless Autoconfiguration, page 24

Table 5 **Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)**

Feature Name	Releases	Feature Information
IPv6: IPv6 stateless autoconfiguration	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The IPv6 stateless autoconfiguration feature can be used to manage link, subnet, and site addressing changes.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Link-Local Address, page 7 • IPv6 Neighbor Solicitation Message, page 19 • IPv6 Router Advertisement Message, page 21 • IPv6 Stateless Autoconfiguration, page 24 • Simplified Network Renumbering for IPv6 Hosts, page 24
IPv6: IPv6 static cache entry for neighbor discovery	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(8)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The IPv6 static cache entry for neighbor discovery feature allows static entries to be made in the IPv6 neighbor cache.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Neighbor Discovery, page 18
IPv6: Base protocols high availability	Cisco IOS XE Release 2.1	<p>IPv6 neighbor discovery supports SSO.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Neighbor Discovery, page 18
IPv6 access services: Remote bridged encapsulation (RBE)	12.3(4)T 12.4 12.4(2)T	<p>RBE provides a mechanism for routing a protocol from a bridged interface to another routed or bridged interface.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • Routed Bridge Encapsulation for IPv6, page 26
IPv6: Anycast Address	12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(4)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>An anycast address is an address that is assigned to a set of interfaces that typically belong to different nodes.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Address Type: Anycast, page 9 • IPv6 Address Type: Multicast, page 10 • IPv6 Multicast Groups, page 28 • Configuring IPv6 Addressing and Enabling IPv6 Routing, page 28

Table 5 **Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)**

Feature Name	Releases	Feature Information
IPv6 address types: Unicast	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>An IPv6 unicast address is an identifier for a single interface, on a single node.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Address Formats, page 4 • IPv6 Address Type: Unicast, page 5 • IPv6 Address Type: Anycast, page 9 • IPv6 Address Type: Multicast, page 10 • IPv6 Neighbor Solicitation Message, page 19 • IPv6 Router Advertisement Message, page 21 • Configuring IPv6 Addressing and Enabling IPv6 Routing, page 28
Unicast Reverse Path Forwarding for IPv6	12.0(31)S Cisco IOS XE Release 2.1	<p>The Unicast RPF feature mitigates problems caused by malformed or forged (spoofed) IPv6 source addresses that pass through an IPv6 router.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Prerequisites for Implementing IPv6 Addressing and Basic Connectivity, page 2 • Unicast Reverse Path Forwarding, page 16
IPv6 data link: ATM PVC and ATM LANE	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. ATM PVC and ATM LANE are data links supported for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36 • Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces, page 42
IPv6 data link: Cisco High-Level Data Link Control (HDLC)	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. HDLC is a type of data link supported for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36 • Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces, page 42

Table 5 **Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)**

Feature Name	Releases	Feature Information
IPv6 data link: Dynamic packet transport (DPT)	12.0(23)S	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. DPT is a type of data link supported for IPv6.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26
IPv6 data link: Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet are data links supported for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36
IPv6 data link: FDDI	12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. FDDI is a type of data link supported for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36
IPv6 data link: Frame Relay PVC	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 networks, a data link is a network sharing a particular link-local prefix. Frame relay PVC is a type of data link supported for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36 • Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces, page 42

Table 5 *Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)*

Feature Name	Releases	Feature Information
IPv6 data link: PPP service over Packet over SONET, ISDN, and serial (synchronous and asynchronous) interfaces	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	In IPv6 networks, a data link is a network sharing a particular link-local prefix. PPP service over Packet over SONET, ISDN, and serial interfaces is a type of data link supported for IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Data Links, page 26 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36 • Mapping IPv6 Addresses to IPv6 ATM and Frame Relay Interfaces, page 42
IPv6 data link: VLANs using Cisco Inter-Switch Link (ISL)	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	In IPv6 networks, a data link is a network sharing a particular link-local prefix. VLANs using Cisco ISL is a type of data link supported for IPv6. The following section provides information about this feature: <ul style="list-style-type: none"> • IPv6 Data Links, page 26
IPv6 data link: VLANs using IEEE 802.1Q encapsulation	12.0(22)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(14)S 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	In IPv6 networks, a data link is a network sharing a particular link-local prefix. VLANs using IEEE 802.1Q encapsulation is a type of data link supported for IPv6. The following section provides information about this feature: <ul style="list-style-type: none"> • IPv6 Data Links, page 26
IPv6 services: AAAA DNS lookups over an IPv4 transport	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IPv6 basic connectivity can be enhanced by configuring support for AAAA record types in the DNS name-to-address and address-to-name lookup processes. The following section provides information about this feature: <ul style="list-style-type: none"> • DNS for IPv6, page 17

Table 5 **Feature Information for Implementing IPv6 Addressing and Basic Connectivity (continued)**

Feature Name	Releases	Feature Information
IPv6 services: Cisco Discovery Protocol—IPv6 address family support for neighbor information	12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(8)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	The Cisco Discovery Protocol IPv6 address support for neighbor information feature adds the ability to transfer IPv6 addressing information between two Cisco devices. The following section provides information about this feature: <ul style="list-style-type: none"> • Cisco Discovery Protocol IPv6 Address Support, page 17
IPv6 services: DNS lookups over an IPv6 transport	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(8)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The following section provides information about this feature: <ul style="list-style-type: none"> • DNS for IPv6, page 17
IPv6 services: generic prefix	12.3(4)T 12.4 12.4(2)T	The upper 64 bits of an IPv6 address are composed from a global routing prefix plus a subnet ID. A general prefix (for example, /48) holds a short prefix, based on which a number of longer, more specific, prefixes (for example, /64) can be defined. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 General Prefixes, page 25 • Defining and Using IPv6 General Prefixes, page 30
IPv6 switching: Cisco Express Forwarding and distributed Cisco Express Forwarding support	12.0(21)ST 12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Cisco Express Forwarding for IPv6 is advanced, Layer 3 IP switching technology for the forwarding of IPv6 packets. Distributed Cisco Express Forwarding for IPv6 performs the same functions as CEFv6 but for distributed architecture platforms such as the Cisco 12000 series Internet routers and the Cisco 7500 series routers. The following sections provide information about this feature: <ul style="list-style-type: none"> • Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 15 • Configuring Cisco Express Forwarding and Distributed Cisco Express Forwarding Switching for IPv6, page 36

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing ADSL and Deploying Dial Access for IPv6

First Published: November 25, 2002

Last Updated: May 1, 2006

This module describes the implementation of prefix pools and per-user Remote Access Dial-In User Service (RADIUS) attributes in IPv6. It also describes the deployment of IPv6 in Digital Subscriber Line (DSL) and dial-access environments. Asymmetric Digital Subscriber Line (ADSL) and dial deployment provide the extensions that make large-scale access possible for IPv6 environments, including IPv6 RADIUS attributes, stateless address configuration on Point-to-Point Protocol (PPP) links, per-user static routes, and access control lists (ACLs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing ADSL and Deploying Dial Access for IPv6” section on page 18](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing ADSL and Dial Access for IPv6, page 2](#)
- [Restrictions for Implementing ADSL and Deploying Dial Access for IPv6, page 2](#)
- [Information About Implementing ADSL and Deploying Dial Access for IPv6, page 2](#)
- [How to Configure ADSL and Deploy Dial Access in IPv6, page 6](#)
- [Configuration Examples for Implementing ADSL and Deploying Dial Access for IPv6, page 13](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2002–2009 Cisco Systems, Inc. All rights reserved.

- [Where to Go Next, page 14](#)
- [Additional References, page 15](#)
- [Command Reference, page 16](#)
- [Feature Information for Implementing ADSL and Deploying Dial Access for IPv6, page 18](#)

Prerequisites for Implementing ADSL and Dial Access for IPv6

This document assumes that you are familiar with IPv4. Refer to the publications referenced in the [“Additional References”](#) section for IPv4 configuration and command reference information.

Restrictions for Implementing ADSL and Deploying Dial Access for IPv6

ADSL and Dial Deployment is available for interfaces with PPP encapsulation enabled, including PPP over ATM (PPPoA), PPP over Ethernet (PPPoE), PPP over async, and PPP over ISDN.

Information About Implementing ADSL and Deploying Dial Access for IPv6

To implement ADSL and deploy dial access for IPv6, you need to understand the following concepts:

- [Address Assignment for IPv6, page 2](#)
- [AAA Attributes for IPv6, page 3](#)

Address Assignment for IPv6

A Cisco router configured with IPv6 will advertise its IPv6 prefixes on one or more interfaces, allowing IPv6 clients to automatically configure their addresses. In IPv6, address assignment is performed at the network layer, in contrast to IPv4 where a number of functions are handled in the PPP layer. The only function handled in IPv6 Control Protocol (IPv6CP) is the negotiation of a unique interface identifier. Everything else, including DNS server discovery, is done within the IPv6 protocol itself.

Contrary to IPv4 address assignment, an IPv6 user will be assigned a prefix, not a single address. Typically the Internet Service Provider (ISP) assigns a 64- or 48-bit prefix.

In the IPv6 world, Internet service providers (ISPs) assign long-lived prefixes to users, which has some impact on the routing system. In typical IPv4 environments, each network access server (NAS) has a pool of 24-bit addresses and users get addresses from this pool when dialing in. If a user dials another POP or is connected to another NAS at the same POP, a different IPv4 address is assigned.

Addresses for IPv6 are assigned by two different methods.

- [Stateless Address Autoconfiguration, page 3](#)
- [Prefix Delegation, page 3](#)

Stateless Address Autoconfiguration

Assigning addresses using the stateless address autoconfiguration method can only be used to assign 64-bit prefixes. Each user is assigned a 64-bit prefix, which is advertised to the user in a router advertisement (RA). All addresses are automatically configured based on the assigned prefix.

A typical scenario is to assign a separate 64-bit prefix per user; however, users can also be assigned a prefix from a shared pool of addresses. Using the shared limits addresses to only one address per user.

This solution works best for the cases where the customer provider edge router (CPE) is a single PC or is limited to only one subnet. If the user has multiple subnets, Layer 2 (L2) bridging, multilink subnets or proxy RA can be used. The prefix advertised in the RA can come from an authorization, authentication, and accounting (AAA) server, which also provides the prefix attribute, can be manually configured, or can be allocated from a prefix pool.

The Framed-Interface-Id AAA attribute influences the choice of interface identifier for peers and, in combination with the prefix, the complete IPv6 address can be determined.

Prefix Delegation

Prefix delegation uses Dynamic Host Configuration Protocol (DHCP). When the user requests a prefix from the prefix delegator, typically the NAS, the prefix is allocated as described in the [“Stateless Address Autoconfiguration” section on page 3](#).

An IPv6 prefix delegating router selects IPv6 prefixes to be assigned to a requesting router upon receiving a request from the client. The delegating router might select prefixes for a requesting router in the following ways:

- Static assignment based on subscription to an ISP
- Dynamic assignment from a pool of available prefixes
- Selection based on an external authority such as a RADIUS server using the Framed-IPv6-Prefix attribute (see the [“Framed-IPv6-Prefix” section on page 4](#)).

DHCP SIP Server Options

Two DHCP for IPv6 Session Initiation Protocol (SIP) server options describe a local outbound SIP proxy: one carries a list of domain names, the other a list of IPv6 addresses. These two options can be configured in a DHCPv6 configuration pool.

AAA Attributes for IPv6

Vendor-specific attributes (VSAs) have been developed to support AAA for IPv6. The Cisco VSAs are `inac1`, `outac1`, `route`, and `prefix`.

Prefix pools and pool names are configurable through AAA.

The following RADIUS attributes as described in RFC 3162 are supported for IPv6:

- Framed-Interface-Id
- Framed-IPv6-Prefix
- Login-IPv6-Host
- Framed-IPv6-Route
- Framed-IPv6-Pool

These attributes can be configured on a RADIUS server and downloaded to access servers where they can be applied to access connections.

AAA attributes are described in the following sections:

- [RADIUS Per-User Attributes for Virtual Access in IPv6 Environments, page 4](#)
- [IPv6 Prefix Pools, page 6](#)

Prerequisites for Using AAA Attributes for IPv6

The AAA attributes for IPv6 are compliant with RFC 3162 and require a RADIUS server capable of supporting RFC 3162.

RADIUS Per-User Attributes for Virtual Access in IPv6 Environments

The following IPv6 attributes for RADIUS attribute-value (AV) pairs are supported for virtual access:

- [Framed-Interface-Id, page 4](#)
- [Framed-IPv6-Prefix, page 4](#)
- [Login-IPv6-Host, page 5](#)
- [Framed-IPv6-Route, page 5](#)
- [Framed-IPv6-Pool, page 5](#)
- [IPv6 Route, page 5](#)
- [IPv6 ACL, page 5](#)
- [IPv6 Prefix#, page 5](#)
- [IPv6 Pool, page 5](#)

Apart from the new IPv6 prefix and IPv6 pool attributes, these are all existing Cisco VSAs extended to support the IPv6 protocol.

Framed-Interface-Id

The Framed-Interface-Id attribute indicates the IPv6 interface identifier to be configured. This per-user attribute is used during the IPv6CP negotiations and may be used in access-accept packets. If the Interface-Identifier IPv6CP option has been successfully negotiated, this attribute must be included in an Acc-0Request packet as a hint by the NAS to the server that it would prefer that value.

Framed-IPv6-Prefix

The Framed-IPv6-Prefix attribute performs the same function as the Cisco VSA: It is used for virtual access only and indicates an IPv6 prefix (and corresponding route) to be configured. This attribute is a per-user attribute and lets the user specify which prefixes to advertise in Neighbor Discovery Router Advertisement messages. The Framed-IPv6-Prefix attribute may be used in access-accept packets and can appear multiple times. The NAS will create a corresponding route for the prefix.

To use this attribute for DHCP for IPv6 prefix delegation, create a profile for the same user on the RADIUS server. The user name associated with the second profile has the suffix “-dhcipv6.”

The Framed-IPv6-Prefix attribute in the two profiles is treated differently. If a NAS needs both to send a prefix in router advertisements (RAs) and delegate a prefix to a remote user's network, the prefix for RA is placed in the Framed-IPv6-Prefix attribute in the user's regular profile, and the prefix used for prefix delegation is placed in the attribute in the user's separate profile.

Login-IPv6-Host

The Login-IPv6-Host attribute is a per-user attribute that indicates the IPv6 system with which to connect the user when the Login-Service attribute is included.

Framed-IPv6-Route

The Framed-IPv6-Route attribute performs the same function as the Cisco VSA: It is a per-user attribute that provides routing information to be configured for the user on the NAS. This attribute is a string attribute and is specified using the **ipv6 route** command.

Framed-IPv6-Pool

The IPv6-Pool attribute is a per-user attribute that contains the name of an assigned pool that should be used to assign an IPv6 prefix for the user. This pool should either be defined locally on the router or defined on a RADIUS server from which pools can be downloaded.

IPv6 Route

The IPv6 route attribute allows you to specify a per-user static route. A static route is appropriate when the Cisco IOS software cannot dynamically build a route to the destination. See the description of the **ipv6 route** command for more information about building static routes.

The following example shows the IPv6 route attribute used to define a static route:

```
cisco-avpair = "ipv6:route#1=2001:0DB8:cc00:1::/48",  
cisco-avpair = "ipv6:route#2=2001::0DB8:cc00:2::/48",
```

IPv6 ACL

You can specify a complete IPv6 access list. The unique name of the access list is generated automatically. The access list is removed when its user logs out. The previous access list on the interface is reapplied.

The **inacl** and **outacl** attributes allow you to a specific existing access list configured on the router. The following example shows ACL number 1 specified as the access list:

```
cisco-avpair = "ipv6:inacl#1=permit 2001:0DB8:cc00:1::/48",  
cisco-avpair = "ipv6:outacl#1=deny 2001:0DB8::/10",
```

IPv6 Prefix#

The IPv6 prefix# attribute lets you indicate which prefixes to advertise in Neighbor Discovery Router Advertisement messages. When the prefix# attribute is used, a corresponding route (marked as a per-user static route) is installed in the routing information base (RIB) tables for the given prefix.

```
cisco-avpair = "ipv6:prefix#1=2001:0db8:/64",  
cisco-avpair = "ipv6:prefix#2=2001:0db8:/64",
```

IPv6 Pool

For RADIUS authentication, the IPv6 pool attribute extends the IPv4 address pool attributed to support the IPv6 protocol. It specifies the name of a local pool on the NAS from which to get the prefix and is used whenever the service is configured as PPP and whenever the protocol is specified as IPv6. Note that the address pool works in conjunction with local pooling. It specifies the name of the local pool that has been preconfigured on the NAS.

IPv6 Prefix Pools

The function of prefix pools in IPv6 is similar to that of address pools in IPv4. The main difference is that IPv6 assigns prefixes rather than single addresses.

As for IPv4, a pool or a pool definition can be configured locally or it can be retrieved from an AAA server. Overlapping membership between pools is not permitted.

Once a pool is configured, it cannot be changed. If you change the configuration, the pool will be removed and re-created. All prefixes previously allocated will be freed.

Prefix pools can be defined so that each user is allocated a 64-bit prefix or so that a single prefix is shared among several users. In a shared prefix pool, each user may receive only one address from the pool.

How to Configure ADSL and Deploy Dial Access in IPv6

The configuration guidelines contained in this section show how to configure ADSL and dial access in IPv6 environments.

- [Configuring the NAS, page 6](#) (required)
- [Configuring the Remote CE Router, page 9](#) (required)
- [Configuring the DHCP for IPv6 Server to Obtain Prefixes from RADIUS Servers, page 11](#) (optional)
- [Configuring DHCP for IPv6 AAA and SIP Options, page 12](#) (optional)

Configuring the NAS

The first step in setting up dial access is to configure the NAS. All of the dialer groups, access lists, and routes are known to the NAS. This task shows how to configure the NAS to implement ADSL and deploy dial access for IPv6 environments.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **aaa new-model**
5. **aaa authentication ppp** { **default** | *list-name* } *method1* [*method2...*]
6. **aaa authorization configuration default** { **radius** | **tacacs+** }
7. **show ipv6 route** [*ipv6-address* | *ipv6-prefix/prefix-length* | *protocol* | *interface-type interface-number*]
8. **virtual-profile virtual-template** *number*
9. **interface serial** *controller-number:timeslot*
10. **encapsulation** *encapsulation-type*
11. **exit**
12. **dialer-group** *group-number*

13. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**] [**optional**]
14. **interface virtual-template** *number*
15. **ipv6 enable**
16. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | *access-group*}
17. **radius-server host** {*hostname* | *ip-address*} [**test username** *user-name*] [**auth-port** *port-number*] [**ignore-auth-port**] [**acct-port** *port-number*] [**ignore-acct-port**] [**timeout** *seconds*] [**retransmit** *retries*] [**key string**] [**alias** {*hostname* | *ip-address*}] [**idle-time** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	hostname <i>name</i> Example: Router(config)# hostname cust1-53a	Specifies the host name for the network server.
Step 4	aaa new-model Example: Router(config)# aaa new-model	Enables the AAA server.
Step 5	aaa authentication ppp { default <i>list-name</i> } <i>method1</i> [<i>method2...</i>] Example: Router(config)# aaa authentication ppp default if-needed group radius	Specifies one or more AAA authentication methods for use on serial interfaces that are running PPP.
Step 6	aaa authorization configuration default { radius tacacs+ } Example: Router(config)# aaa authorization network default group radius	Downloads configuration information from the AAA server.

	Command or Action	Purpose
Step 7	show ipv6 route [<i>ipv6-address</i> <i>ipv6-prefix/prefix-length</i> <i>protocol</i> <i>interface-type interface-number</i>] Example: Router(config)# show ipv6 route	Shows the routes installed by the previous commands.
Step 8	virtual-profile virtual-template <i>number</i> Example: Router(config)# virtual-profile virtual-template 1	Enables virtual profiles by virtual interface template.
Step 9	interface serial <i>controller-number:timeslot</i> Example: Router(config)# interface Serial0:15	Specifies a serial interface created on a channelized E1 or channelized T1 controller (for ISDN PRI, channel-associated signaling, or robbed-bit signaling).
Step 10	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 11	exit Example: Router(config-if)# exit	Returns to global configuration mode.
Step 12	dialer-group <i>group-number</i> Example: Router(config)# dialer-group 1	Control access by configuring an interface to belong to a specific dialing group.
Step 13	ppp authentication { <i>protocol1</i> [<i>protocol2...</i>]} [<i>if-needed</i>] [<i>list-name</i> default] [callin] [one-time] [optional] Example: Router(config)# ppp authentication chap	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 14	interface virtual-template <i>number</i> Example: Router(config)# interface virtual-template1	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
Step 15	ipv6 enable Example: Router(config)# ipv6 enable	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 16	dialer-list <i>dialer-group</i> protocol <i>protocol-name</i> { permit deny list <i>access-list-number</i> <i>access-group</i> } Example: Router(config)# dialer-list 1 protocol ipv6 permit	Defines a dial-on-demand routing (DDR) dialer list for dialing by protocol or by a combination of a protocol and a previously defined access list.
Step 17	radius-server host { <i>hostname</i> <i>ip-address</i> } [test username <i>user-name</i>] [auth-port <i>port-number</i>] [ignore-auth-port] [acct-port <i>port-number</i>] [ignore-acct-port] [timeout <i>seconds</i>] [retransmit <i>retries</i>] [key <i>string</i>] [alias { <i>hostname</i> <i>ip-address</i> }] [idle-time <i>seconds</i>] Example: Router(config)# radius-server host 172.17.250.8 auth-port 1812 acct-port 1813 key testing123	Specifies a RADIUS server host.

Troubleshooting Tips

Verify that the access list is installed correctly before proceeding with the next task. Use the **show ipv6 access-list** and **show ipv6 interface** commands.

Configuring the Remote CE Router

The following task describes how to configure each remote CE router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **interface** *bri number.subinterface-number* [**multipoint** | **point-to-point**]
5. **encapsulation** *encapsulation-type*
6. **ipv6 address** **autoconfig** [**default**]
7. **isdn switch-type** *switch-type*
8. **ppp authentication** {*protocol1* [*protocol2...*]} [**if-needed**] [*list-name* | **default**] [**callin**] [**one-time**]
9. **ppp multilink** [**bap** | **required**]
10. **exit**
11. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* | *access-group*}
12. **ipv6 route** *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	hostname <i>name</i> Example: Router(config)# hostname cust1-36a	Specifies the host name for the network server.
Step 4	interface bri <i>number.subinterface-number</i> [multipoint point-to-point] Example: Router(config)# interface BRI1/0	Configures a BRI interface and enters interface configuration mode.
Step 5	encapsulation <i>encapsulation-type</i> Example: Router(config-if)# encapsulation ppp	Sets the encapsulation method used by the interface.
Step 6	ipv6 address autoconfig [default] Example: Router(config-if)# ipv6 address autoconfig	Indicates that the IPv6 address will be generated automatically.
Step 7	isdn switch-type <i>switch-type</i> Example: Router(config-if)# isdn switch-type basic-net3	Specifies the central office switch type on the ISDN interface.
Step 8	ppp authentication { <i>protocol1</i> [<i>protocol2...</i>]} [if-needed] [<i>list-name</i> default] [callin] [one-time] Example: Router(config-if)# ppp authentication chap optional	Enables Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) or both and specifies the order in which CHAP and PAP authentication are selected on the interface.
Step 9	ppp multilink [bap required] Example: Router(config-if)# ppp multilink	Enables Multilink PPP (MLP) on an interface and, optionally, enables Bandwidth Allocation Control Protocol (BACP) and Bandwidth Allocation Protocol (BAP) for dynamic bandwidth allocation.

	Command or Action	Purpose
Step 10	exit Example: Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 11	dialer-list <i>dialer-group</i> protocol <i>protocol-name</i> { permit deny list <i>access-list-number</i> <i>access-group</i> } Example: Router(config)# dialer-list 1 protocol ipv6 permit	Defines a dial-on-demand routing (DDR) dialer list for dialing by protocol or by a combination of a protocol and a previously defined access list.
Step 12	ipv6 route <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>]} [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>tag tag</i>] Example: Router(config)# ipv6 route 2001:0db8:1/128 BRI1/0	Establishes static IPv6 routes. Use one command for each route.

What to Do Next

Once you have configured the NAS and CE router, configure RADIUS to establish the AV pairs for callback. Callback allows remote network users to dial in to the NAS without being charged. When callback is required, the NAS hangs up the current call and dials the caller back. When the NAS performs the callback, only information for the outgoing connection is applied. The rest of the attributes from the preauthentication access-accept message are discarded.

The following example shows a RADIUS profile configuration for a local campus:

```
campus1 Auth-Type = Local, Password = "mypassword"
      User-Service-Type = Framed-User,
      Framed-Protocol = PPP,
      cisco-avpair = "ipv6:inac1#1=permit dead::/64 any",
      cisco-avpair = "ipv6:route=dead::/64",
      cisco-avpair = "ipv6:route=cafe::/64",
      cisco-avpair = "ipv6:prefix=dead::/64 0 0 onlink autoconfig",
      cisco-avpair = "ipv6:prefix=cafe::/64 0 0 onlink autoconfig",
      cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
```

The RADIUS AV pairs for IPv6 are described in [RADIUS Per-User Attributes for Virtual Access in IPv6 Environments, page 4](#).

Configuring the DHCP for IPv6 Server to Obtain Prefixes from RADIUS Servers

The following task describes how to configure the DHCP for IPv6 server to obtain prefixes from RADIUS servers.

Prerequisites

Before you perform this task, you must configure the AAA client and PPP on the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd prefix framed-ipv6-prefix**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 nd prefix framed-ipv6-prefix Example: Router(config-if)# ipv6 nd prefix framed-ipv6-prefix	Adds the prefix in a received RADIUS framed IPv6 prefix attribute to the interface's neighbor discovery prefix queue.

Configuring DHCP for IPv6 AAA and SIP Options

This optional task allows users to enable the router to support AAA and SIP options.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 dhcp pool** *poolname*
4. **prefix-delegation aaa** [**method-list** *method-list*] [*lifetime*]
5. **sip address** *ipv6-address*
6. **sip domain-name** *domain-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 dhcp pool poolname Example: Router(config)# ipv6 dhcp pool pool1	Configures a DHCP for IPv6 configuration information pool and enters DHCP for IPv6 pool configuration mode.
Step 4	prefix-delegation aaa [method-list method-list] [lifetime] Example: Router(config-dhcp)# prefix-delegation aaa method-list list1	Specifies that prefixes are to be acquired from AAA servers.
Step 5	sip address ipv6-address Example: Router(config-dhcp)# sip address 2001:0DB8::2	Configures a SIP server IPv6 address to be returned in the SIP server's IPv6 address list option to clients.
Step 6	sip domain-name domain-name Example: Router(config-dhcp)# sip domain sip1.cisco.com	Configures a SIP server domain name to be returned in the SIP server's domain name list option to clients.

Configuration Examples for Implementing ADSL and Deploying Dial Access for IPv6

This section provides the following configuration example:

- [Implementing ADSL and Deploying Dial Access for IPv6: Example, page 13](#)

Implementing ADSL and Deploying Dial Access for IPv6: Example

This example shows a typical configuration for ADSL and dial access. The following three separate configurations are required:

- [NAS Configuration](#)
- [Remote CE Router Configuration](#)
- [RADIUS Configuration](#)

NAS Configuration

This configuration for the ISP NAS shows the configuration that supports access from the remote CE router.

```
hostname cust1-53a
aaa new-model
aaa authentication ppp default if-needed group radius
aaa authorization network default group radius
virtual-profile virtual-template 1
interface Serial0:15
    encapsulation ppp
    dialer-group 1
    ppp authentication chap
!
interface Virtual-Template1
    ipv6 enable
!
dialer-list 1 protocol ipv6 permit
radius-server host 172.17.250.8 auth-port 1812 acct-port 1813 key testing123
```

Remote CE Router Configuration

This configuration for the remote customer edge router shows PPP encapsulation and IPv6 routes defined.

```
hostname cust-36a
interface BRI1/0
    encapsulation ppp
    ipv6 enable
    isdn switch-type basic-net3
    ppp authentication chap optional
    ppp multilink
!
dialer-list 1 protocol ipv6 permit
ipv6 route 2001:0DB8:1/128 BRI1/0
ipv6 route ::/0 2001:0db8:1
```

RADIUS Configuration

This RADIUS configuration shows the definition of AV pairs to establish the static routes.

```
campus1 Auth-Type = Local, Password = "mypassword"
    User-Service-Type = Framed-User,
    Framed-Protocol = PPP,
    cisco-avpair = "ipv6:inacl#1=permit dead::/64 any",
    cisco-avpair = "ipv6:route=library::/64",
    cisco-avpair = "ipv6:route=cafe::/64",
    cisco-avpair = "ipv6:prefix=library::/64 0 0 onlink autoconfig",
    cisco-avpair = "ipv6:prefix=cafe::/64 0 0 onlink autoconfig",
    cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
```

Where to Go Next

For information about implementing routing protocols for IPv6, refer to the [Implementing RIP for IPv6](#), [Implementing IS-IS for IPv6](#), or the [Implementing Multiprotocol BGP for IPv6](#) module. For information about implementing security for IPv6 environments, refer to the [Implementing IPsec in IPv6 Security](#) and [Implementing Traffic Filters and Firewalls for IPv6 Security](#) modules.

Additional References

The following sections provide references related to the Implementing ADSL and Deploying Dial Access for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 basic connectivity	“Implementing IPv6 Addressing and Basic Connectivity,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
Certification authority and interoperability, RA proxy	“Security Overview,” <i>Cisco IOS Security Configuration Guide</i>
RADIUS server configuration	“Security Overview,” <i>Cisco IOS Security Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3162	<i>RADIUS and IPv6</i>
RFC 3177	<i>IAB/IESG Recommendations on IPv6 Address</i>
RFC 3319	<i>Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiated Protocol (SIP) Servers</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **aaa new-model**
- **debug ipv6 pool**
- **dialer-list protocol**
- **hostname**
- **ipv6 dhcp pool**
- **ipv6 local pool**
- **ipv6 nd prefix framed-ipv6-prefix**
- **ipv6 route**
- **peer default ipv6 address pool**
- **prefix-delegation aaa**
- **show ipv6 local pool**

- **show ipv6 route**
- **sip address**
- **sip domain-name**

Feature Information for Implementing ADSL and Deploying Dial Access for IPv6

Table 15 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(13)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 15 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 15 identifies the earliest release for each early-deployment train in which each feature became available.

Table 15 Feature Information for Implementing ADSL and Deploying Dial Access for IPv6

Feature Name	Releases	Feature Information
Enhanced IPv6 features for ADSL and dial deployment	12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	Several features were enhanced to enable IPv6 to use ADSL and dial deployment. The following sections provide information about these features: <ul style="list-style-type: none"> • Address Assignment for IPv6, page 2 • Stateless Address Autoconfiguration, page 3 • Configuring the NAS, page 6 • Configuring the Remote CE Router, page 9
AAA support for Cisco VSA IPv6 attributes	12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	Vendor-specific attributes (VSAs) were developed to support AAA for IPv6. The following section provides information about this feature: <ul style="list-style-type: none"> • AAA Attributes for IPv6, page 3

Table 15 *Feature Information for Implementing ADSL and Deploying Dial Access for IPv6 (continued)*

Feature Name	Releases	Feature Information
PPPoA	12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	ADSL and dial deployment is available for interfaces with PPP encapsulation enabled, including PPPoA. The following sections provide information about these features: <ul style="list-style-type: none"> • Address Assignment for IPv6, page 2 • Configuring the NAS, page 6 • Configuring the Remote CE Router, page 9
PPPoE	12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	ADSL and dial deployment is available for interfaces with PPP encapsulation enabled, including PPPoE. The following sections provide information about these features: <ul style="list-style-type: none"> • Address Assignment for IPv6, page 2 • Configuring the NAS, page 6 • Configuring the Remote CE Router, page 9
IPv6 prefix pools	12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T	The function of prefix pools in IPv6 is similar to that of address pools in IPv4. The main difference is that IPv6 assigns prefixes rather than single addresses. The following sections provide information about these features: <ul style="list-style-type: none"> • AAA Attributes for IPv6, page 3 • Stateless Address Autoconfiguration, page 3 • IPv6 Prefix Pools, page 6 • Configuring the NAS, page 6 • Configuring the Remote CE Router, page 9
AAA support for RFC 3162 IPv6 RADIUS attributes	12.3(4)T 12.4 12.4(2)T	The AAA attributes for IPv6 are compliant with RFC 3162 and require a RADIUS server capable of supporting RFC 3162. The following sections provide information about these features: <ul style="list-style-type: none"> • AAA Attributes for IPv6, page 3 • Prerequisites for Using AAA Attributes for IPv6, page 4 • RADIUS Per-User Attributes for Virtual Access in IPv6 Environments, page 4 • Configuring the DHCP for IPv6 Server to Obtain Prefixes from RADIUS Servers, page 11 • Configuring DHCP for IPv6 AAA and SIP Options, page 12

Table 15 *Feature Information for Implementing ADSL and Deploying Dial Access for IPv6 (continued)*

Feature Name	Releases	Feature Information
DHCPv6 prefix delegation	12.0(32)S 12.2(28)SB 12.2(33)SRA 12.2(18)SXE 12.3(4)T 12.4 12.4(2)T	When the user requests a prefix from the prefix delegator, typically the NAS, the prefix is allocated using DHCPv6. The following sections provide information about these features: <ul style="list-style-type: none"> • Prefix Delegation, page 3 • Configuring the DHCP for IPv6 Server to Obtain Prefixes from RADIUS Servers, page 11
DHCP for IPv6 prefix delegation via AAA	12.2(18)SXE 12.3(14)T 12.4 12.4(2)T	The following sections provide information about these features: <ul style="list-style-type: none"> • Stateless Address Autoconfiguration, page 3 • Prefix Delegation, page 3 • IPv6 Prefix Pools, page 6 • Prerequisites for Using AAA Attributes for IPv6, page 4 • Configuring DHCP for IPv6 AAA and SIP Options, page 12

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2002–2009 Cisco Systems, Inc. All rights reserved.



Implementing Bidirectional Forwarding Detection for IPv6

First Published: May 5, 2008

Last Updated: August 18, 2008

This document describes how to implement the Bidirectional Forwarding Detection for IPv6 (BFDv6) protocol. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. BFDv6 provides IPv6 support by accommodating IPv6 addresses, and it provides the ability to create BFDv6 sessions.

Because the network administrator can use BFD to detect forwarding path failures at a uniform rate, rather than the variable rates for different routing protocol hello mechanisms, network profiling and planning will be easier, and convergence time will be consistent and predictable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Bidirectional Forwarding for IPv6” section on page 15](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing Bidirectional Forwarding for IPv6, page 2](#)
- [Restrictions for Implementing Bidirectional Forwarding for IPv6, page 2](#)
- [Information About Implementing Bidirectional Forwarding for IPv6, page 2](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2008–2009 Cisco Systems, Inc. All rights reserved.

- [How to Configure Bidirectional Forwarding Detection for IPv6, page 5](#)
- [Configuration Examples for Bidirectional Forwarding Detection for IPv6, page 12](#)
- [Additional References, page 12](#)
- [Command Reference, page 13](#)
- [Feature Information for Implementing Bidirectional Forwarding for IPv6, page 15](#)

Prerequisites for Implementing Bidirectional Forwarding for IPv6

Cisco Express Forwarding and IPv6 unicast routing must be enabled on all participating routers.

Restrictions for Implementing Bidirectional Forwarding for IPv6

- BFDv6 supports only global IPv6 neighbor addresses if a global IPv6 address is configured on the interface.
- Only asynchronous mode is supported. In asynchronous mode, either BFDv6 peer can initiate a BFDv6 session.

Information About Implementing Bidirectional Forwarding for IPv6

To configure BFD for IPv6, you need to understand the following concepts:

- [Overview of the BFDv6 Protocol, page 2](#)
- [Static Route Support for BFD over IPv6, page 3](#)
- [BFD Support for OSPFv3, page 4](#)

Overview of the BFDv6 Protocol

This section describes the BFDv6 protocol, how it is different from BFD for IPv4, and how it works with BFD for IPv4. BFD is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols. In addition to fast forwarding path failure detection, BFD provides a consistent failure detection method for network administrators. BFDv6 provides IPv6 support by accommodating IPv6 addresses and provides the ability to create BFDv6 sessions.

BFDv6 Registration

BFD clients register with BFD using a registry application program interface (API). The registry arguments include protocol type and the address and interface description block (IDB) of the route to be monitored. These APIs and arguments are all assumed by BFD to be IPv4.

BFDv6 has registries from which these arguments have been removed, and the protocol and encapsulation are described within a session information structure. These session information structures are defined by BFDv6 for the protocols supported. BFDv6 uses information from the session information structures to determine the correct encapsulation for BFDv6 packets on that session.

BFDv6 Global and Link-Local Addresses

BFDv6 supports both global and link-local IPv6 addresses for neighbor creation. BFDv6 sessions select source addresses to match the neighbor address types (for example, global IPv6 address neighbors must be paired with global IPv6 source addresses and link-local IPv6 address neighbors must be paired with link-local IPv6 source addresses). [Table 1](#) shows the address pairings that BFDv6 supports.

Table 1 *BFDv6 Address Pairings for Neighbor Creation*

Source Address	Destination Address	Status
Global	Global	Supported
Global	Link local	Not supported
Link local	Global	Not supported
Link local	Link local	Supported

Because all IPv6-enabled interfaces have a link-local address and BFDv6 selects the source address, link-local address neighbors are always paired with a link-local interface address. The link-local source address with global destination address is not supported by Cisco Express Forwarding. Therefore, a global IPv6 address must be configured on an interface before a session with a global address neighbor may be established in BFDv6. BFDv6 rejects any sessions in which the neighbor address is global and no global address is configured on the interface.



Note

The behavior of a unique local address (ULA) in BFDv6 is the same as a global address.

BFD for IPv4 and IPv6 on the Same Interface

BFD supports multiple IPv4 and IPv6 sessions per interface, with no restriction on the protocol of those sessions.

Static Route Support for BFD over IPv6

Using the BFDv6 protocol to reach the static route next hop ensures that an IPv6 static route is inserted only in the IPv6 Routing Information Base (RIB) when the next-hop neighbor is reachable. Using the BFDv6 protocol also can remove the IPv6 static route from the IPv6 RIB when the next hop becomes unreachable.

A user can configure IPv6 static BFDv6 neighbors. These neighbor can operate in one of two modes: associated (which is the default) and unassociated. A neighbor can be transitioned between the two modes without interrupting the BFDv6 session associated with the neighbor.

BFDv6 Associated Mode

In BFDv6 associated mode, an IPv6 static route is automatically associated with an IPv6 static BFDv6 neighbor if the static route next hop exactly matches the static BFDv6 neighbor.

An IPv6 static route requests a BFDv6 session for each static BFDv6 neighbor that has one or more associated IPv6 static routes and is configured over an interface on which BFD has been configured. The state of the BFDv6 session will be used to determine whether the associated IPv6 static routes are inserted in the IPv6 RIB. For example, static routes are inserted in the IPv6 RIB only if the BFDv6 neighbor is reachable, and the static route is removed from the IPv6 RIB if the BFDv6 neighbor subsequently becomes unreachable.

BFDv6 associated mode requires a user to configure a BFD neighbor and static route on both the router on which the BFD-monitored static route is required and on the neighboring router.

BFDv6 Unassociated Mode

An IPv6 static BFD neighbor may be configured as unassociated. In this mode, the neighbor is not associated with static routes, and the neighbor always requests a BFDv6 session if the interface has been configured for BFDv6.

Unassociated mode is useful in the following situations:

- Bringing up a BFDv6 session in the absence of an IPv6 static route—This case occurs when a static route is on router A, with router B as the next hop. Associated mode requires users to create both a static BFD neighbor and static route on both routers in order to bring up the BFDv6 session from B to A. Specifying the static BFD neighbor in unassociated mode on router B avoids the need to configure an unwanted static route.
- Transition to BFD monitoring of a static route—This case occurs when existing IPv6 static routes are inserted in the IPv6 RIB. The user wants to enable BFD monitoring for these static routes without any interruption to traffic. If the user configures an attached IPv6 static BFD neighbor, then the static routes will immediately be associated with the new static BFD neighbor. However, because a static BFD neighbor starts in a down state, the associated static routes are then removed from the IPv6 RIB and are reinserted when the BFDv6 session comes up. Therefore, the user will see an interruption in traffic. This interruption can be avoided by configuring the static BFD neighbor as unassociated, waiting until the BFDv6 session has come up, and then reconfiguring the static BFD neighbor as associated.
- Transition from BFD monitoring of a static route—In this case, IPv6 static routes are monitored by BFD and inserted in the RIB. The user wants to disable BFD monitoring of the static routes without interrupting traffic flow. This scenario can be achieved by first reconfiguring the static BFD neighbor as detached (thus disassociating the neighbor from the static routes) and then deconfiguring the static BFD neighbor.

BFD Support for OSPFv3

BFD supports the dynamic routing protocol OSPF for IPv6 (OSPFv3). For information on how to configure OSPFv3, see the [“Configuring BFD Support for OSPFv3”](#) section.

How to Configure Bidirectional Forwarding Detection for IPv6

BFDv6 is configured in much the same way as BFD for IPv4. For information how to configure BFDv6, see “[Bidirectional Forwarding Detection](#)” in the *Cisco IOS IP Routing Protocols Configuration Guide*.

This section describes the following BFDv6 tasks:

- [Specifying a Static BFDv6 Neighbor, page 5](#)
- [Associating an IPv6 Static Route with a BFDv6 Neighbor, page 6](#)
- [Configuring BFD Support for OSPFv3, page 7](#)
- [Retrieving BFDv6 Information for Monitoring and Troubleshooting, page 11](#)

Specifying a Static BFDv6 Neighbor

An IPv6 static BFDv6 neighbor is specified separately from an IPv6 static route. An IPv6 static BFDv6 neighbor must be fully configured with the interface and neighbor address and must be directly attached to the local router.

This task describes how to specify a static BFDv6 neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd [vrf vrf-name] interface-type interface-number ipv6-address [unassociated]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 route static bfd [vrf vrf-name] interface-type interface-number ipv6-address [unassociated]	Specifies static route IPv6 BFDv6 neighbors.
	Example: Router(config)# ipv6 route static bfd ethernet 0/0 2001::1	

Associating an IPv6 Static Route with a BFDv6 Neighbor

IPv6 static routes are automatically associated with a static BFDv6 neighbor. A static neighbor is associated with a BFDv6 neighbor if the static next-hop explicitly matches the BFDv6 neighbor.

This task describes how to associate an IPv6 static route with a BFDv6 neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route static bfd** [*vrf vrf-name*] *interface-type interface-number ipv6-address* [**unassociated**]
4. **ipv6 route** [*vrf vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number [ipv6-address]*} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag tag**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route static bfd [<i>vrf vrf-name</i>] <i>interface-type interface-number ipv6-address</i> [unassociated] Example: Router(config)# ipv6 route static bfd ethernet 0/0 2001::1	Specifies static route BFDv6 neighbors.
Step 4	ipv6 route [<i>vrf vrf-name</i>] <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number [ipv6-address]</i> } [nexthop-vrf [<i>vrf-name1</i> default]] [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>next-hop-address</i>] [tag tag] Example: Router(config)# ipv6 route 2001:0DB8::/64 ethernet 0/0 2001::1	Establishes static IPv6 routes.

Configuring BFD Support for OSPFv3

This section describes the procedures for configuring BFD support for OSPFv3, so that OSPFv3 is a registered protocol with BFD and will receive forwarding path detection failure messages from BFD. You can either configure BFD support for OSPFv3 globally on all interfaces or configure it selectively on one or more interfaces.

There are two methods for enabling BFD support for OSPFv3:

- You can enable BFD for all of the interfaces for which OSPFv3 is routing by using the **bfd all-interfaces** command in router configuration mode. You can disable BFD support on individual interfaces using the **ipv6 ospf bfd disable** command in interface configuration mode.
- You can enable BFD for a subset of the interfaces for which OSPFv3 is routing by using the **ipv6 ospf bfd** command in interface configuration mode.

See the following sections for tasks for configuring BFD support for OSPFv3:

- [Configuring BFD Session Parameters on the Interface, page 7](#)
- [Configuring BFD Support for OSPFv3 for All Interfaces, page 8](#)
- [Configuring BFD Support for OSPFv3 for One or More Interfaces, page 9](#)

Configuring BFD Session Parameters on the Interface

The steps in this procedure show how to configure BFD on the interface by setting the baseline BFD session parameters on an interface. Repeat the steps in this procedure for each interface over which you want to run BFD sessions to BFD neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	bfd interval <i>milliseconds</i> min_rx <i>milliseconds</i> multiplier <i>interval-multiplier</i> Example: Router(config-if)# bfd interval 50 min_rx 50 multiplier 5	Enables BFD on the interface.

Configuring BFD Support for OSPFv3 for All Interfaces

To configure BFD for all OSPFv3 interfaces, perform the steps in this section.

If you do not want to configure BFD on all OSPFv3 interfaces and would rather configure BFD support specifically for one or more interfaces, see the [“Configuring BFD Support for OSPFv3 for One or More Interfaces”](#) section.

Prerequisites

OSPFv3 must be running on all participating routers. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface”](#) section for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id* [**vrf** *vpn-name*]
4. **bfd all-interfaces**
5. **exit** (enter this command twice)
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router ospf <i>process-id</i> [vrf <i>vpn-name</i>] Example: Router(config)# ipv6 router ospf 2	Configures an OSPFv3 routing process.
Step 4	bfd all-interfaces Example: Router(config-router)# bfd all-interfaces	Enables BFD for all interfaces participating in the routing process.
Step 5	exit Example: Router(config-router)# exit	Enter this command twice to go to privileged EXEC mode.
Step 6	show bfd neighbors [vrf <i>vrf-name</i>] [client { bgp eigrp isis ospf rsvp te-frr }] [<i>ip-address</i> ipv6 <i>ipv6-address</i>] [details] Example: Router# show bfd neighbors detail	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] [rate-limit] Example: Router# show ipv6 ospf	(Optional) Displays general information about OSPFv3 routing processes.

Configuring BFD Support for OSPFv3 for One or More Interfaces

To configure BFD on one or more OSPFv3 interfaces, perform the steps in this section.

Prerequisites

OSPFv3 must be running on all participating routers. The baseline parameters for BFD sessions on the interfaces over which you want to run BFD sessions to BFD neighbors must be configured. See the [“Configuring BFD Session Parameters on the Interface”](#) section for more information.

SUMMARY STEPS

1. enable

2. **configure terminal**
3. **interface** *type number*
4. **ipv6 ospf bfd** [**disable**]
5. **exit**
6. **show bfd neighbors** [**vrf** *vrf-name*] [**client** {**bgp** | **eigrp** | **isis** | **ospf** | **rsvp** | **te-frr**}] [*ip-address* | **ipv6** *ipv6-address*] [**details**]
7. **show ipv6 ospf** [*process-id*] [*area-id*] [**rate-limit**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 ospf bfd [disable] Example: Router(config-if)# ipv6 ospf bfd	Enables BFD on a per-interface basis for one or more interfaces associated with the OSPFv3 routing process.
Step 5	exit Example: Router(config-router)# exit	Enter this command twice to go to privileged EXEC mode.
Step 6	show bfd neighbors [vrf <i>vrf-name</i>] [client { bgp eigrp isis ospf rsvp te-frr }] [<i>ip-address</i> ipv6 <i>ipv6-address</i>] [details] Example: Router# show bfd neighbors detail	(Optional) Displays a line-by-line listing of existing BFD adjacencies.
Step 7	show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] [rate-limit] Example: Router# show ipv6 ospf	(Optional) Displays general information about OSPFv3 routing processes.

Retrieving BFDv6 Information for Monitoring and Troubleshooting

This section describes how to retrieve BFDv6 information for maintenance and troubleshooting. The commands in this task can be entered as needed, in any order desired.

SUMMARY STEPS

1. **enable**
2. **monitor event ipv6 static** [enable | disable]
3. **show ipv6 static** [ipv6-address | ipv6-prefix/prefix-length] [interface type number | recursive] [vrf vrf-name] [bfd] [detail]
4. **show ipv6 static** [ipv6-address | ipv6-prefix/prefix-length] [interface type number | recursive] [vrf vrf-name] [bfd] [detail]
5. **debug ipv6 static**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	monitor event ipv6 static [enable disable] Example: Router# monitor event ipv6 static enable	Enables the use of event trace to monitor the operation of the IPv6 static and IPv6 static BFDv6 neighbors.
Step 3	show ipv6 static [ipv6-address ipv6-prefix/prefix-length] [interface type number recursive] [vrf vrf-name] [bfd] [detail] Example: Router# show ipv6 static vrf vrf1 detail	Displays the BFDv6 status for a static route associated with a static BFDv6 neighbor.
Step 4	show ipv6 static [ipv6-address ipv6-prefix/prefix-length] [interface type number recursive] [vrf vrf-name] [bfd] [detail] Example: Router# show ipv6 static vrf vrf1 bfd	Displays static BFDv6 neighbors and associated static routes.
Step 5	debug ipv6 static Example: Router# debug ipv6 static	Enables BFDv6 debugging.

Configuration Examples for Bidirectional Forwarding Detection for IPv6

This section provides the following BFD configuration examples:

- [Specifying an IPv6 Static BFDv6 Neighbor: Example, page 12](#)
- [Associating an IPv6 Static Route with a BFDv6 Neighbor: Example, page 12](#)

Specifying an IPv6 Static BFDv6 Neighbor: Example

The following example specifies a fully configured IPv6 static BFDv6 neighbor. The interface is Ethernet 0/0 and the neighbor address is 2001::1.

```
Router(config)# ipv6 route static bfd ethernet 0/0 2001::1
```

Associating an IPv6 Static Route with a BFDv6 Neighbor: Example

In this example, the IPv6 static route 2001:0DB8::/32 is associated with the BFDv6 neighbor 2001::1 over the Ethernet 0/0 interface:

```
Router(config)# ipv6 route static bfd ethernet 0/0 2001::1
Router(config)# ipv6 route 2001:0DB8::/32 ethernet 0/0 2001::1
```

Additional References

The following sections provide references related to the Bidirectional Forwarding Detection for IPv6 feature.

Related Documents

Related Topic	Document Title
OSPF for IPv6	“Implementing OSPF for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 static routes	“Implementing Static Routes for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
BFD for IPv4	“Bidirectional Forwarding Detection,” Cisco IOS IP Routing Protocols Configuration Guide
BFD for IPv4 commands	“IP Routing Protocol-Independent Commands,” Cisco IOS IP Routing Protocols Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
draft-ietf-bfd-v4v6-1hop-07.txt	<i>BFD for IPv4 and IPv6 (Single Hop)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **bfd all-interfaces**
- **bfd interval**
- **debug ipv6 static**
- **ipv6 ospf bfd**
- **ipv6 route**
- **ipv6 route static bfd**
- **ipv6 router ospf**
- **monitor event ipv6 static**
- **show bfd neighbors**
- **show ipv6 static**

Feature Information for Implementing Bidirectional Forwarding for IPv6

Table 2 lists the release history for this feature.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Implementing Bidirectional Forwarding for IPv6

Feature Name	Releases	Feature Information
IPv6 Routing: Static Route Support for BFD over IPv6	Cisco IOS XE Release 2.1	<p>BFD for IPv6 is used to verify next-hop reachability for IPv6 static routes.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Static Route Support for BFD over IPv6, page 3 • Specifying a Static BFDv6 Neighbor, page 5 • Associating an IPv6 Static Route with a BFDv6 Neighbor, page 6 • Retrieving BFDv6 Information for Monitoring and Troubleshooting, page 11 • Specifying an IPv6 Static BFDv6 Neighbor: Example, page 12 • Associating an IPv6 Static Route with a BFDv6 Neighbor: Example, page 12
OSPFv3 for BFD	Cisco IOS XE Release 2.1	<p>BFD supports the dynamic routing protocol OSPF for IPv6 (OSPFv3).</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • BFD Support for OSPFv3, page 4 • Configuring BFD Support for OSPFv3, page 7

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008–2009 Cisco Systems, Inc. All rights reserved.



Implementing Multiprotocol BGP for IPv6

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure multiprotocol Border Gateway Protocol (BGP) for IPv6. BGP is an Exterior Gateway Protocol (EGP) used mainly to connect separate routing domains that contain independent routing policies (autonomous systems). Connecting to a service provider for access to the Internet is a common use for BGP. BGP can also be used within an autonomous system and this variation is referred to as internal BGP (iBGP). Multiprotocol BGP is an enhanced BGP that carries routing information for multiple network layer protocol address families, for example, IPv6 address family and for IP multicast routes. All BGP commands and routing policy capabilities can be used with multiprotocol BGP.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Multiprotocol BGP for IPv6” section on page 36](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Prerequisites for Implementing Multiprotocol BGP for IPv6, page 2](#)
- [Information About Implementing Multiprotocol BGP for IPv6, page 2](#)
- [How to Implement Multiprotocol BGP for IPv6, page 3](#)
- [Configuration Examples for Multiprotocol BGP for IPv6, page 29](#)
- [Where to Go Next, page 32](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2001–2009 Cisco Systems, Inc. All rights reserved.

- [Additional References, page 32](#)
- [Command Reference, page 34](#)
- [Feature Information for Implementing Multiprotocol BGP for IPv6, page 36](#)

Prerequisites for Implementing Multiprotocol BGP for IPv6

- This module assumes that you are familiar with IPv6 addressing and basic configuration. Refer to the *Implementing IPv6 Addressing and Basic Connectivity* module for more information.
- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for IPv4 configuration and command reference information.
- IPv6 supports the following BGP address family identifiers (AFIs) and subsequent address family identifiers (SAFIs). For further information on AFI and SAFIs, see the *Cisco IOS BGP Configuration Guide*, Release 12.4T:
Cisco IOS XE Release 2.1—IPv6 BGP unicast, IPv6 BGP multicast, VPNv6
- VPN for IPv6 (VPNv6) is supported through IPv6 VPN over MPLS (6VPE)

Information About Implementing Multiprotocol BGP for IPv6

To configure multiprotocol BGP extensions for IPv6, you need to understand the following concepts:

- [Multiprotocol BGP Extensions for IPv6, page 2](#)
- [Multiprotocol BGP for the IPv6 Multicast Address Family, page 2](#)

Multiprotocol BGP Extensions for IPv6

Multiprotocol BGP is the supported EGP for IPv6. Multiprotocol BGP extensions for IPv6 supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multiprotocol BGP include support for an IPv6 address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multiprotocol BGP for the IPv6 Multicast Address Family

The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multiprotocol BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for inter-domain transport of the same. Users must use multiprotocol BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality is provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages. SAFI 1 messages indicate that the routes are only usable for IP unicast, but not IP multicast. Because of this functionality, BGP routes in the IPv6 unicast RIB must be ignored in the IPv6 multicast RPF lookup.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

Nonstop Forwarding and Graceful Restart for MP-BGP IPv6 Address Family

The graceful restart capability is supported for IPv6 BGP unicast, multicast, and VPNv6 address families, enabling Cisco nonstop forwarding (NSF) functionality for BGP IPv6. The BGP graceful restart capability allows the BGP routing table to be recovered from peers without keeping the TCP state.

NSF continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. Forwarding is maintained by synchronizing the FIB between the active and standby RP. On switchover, forwarding is maintained using the FIB. The RIB is not kept synchronized; therefore, the RIB is empty on switchover. The RIB is repopulated by the routing protocols and subsequently informs FIB about RIB convergence by using the NSF_RIB_CONVERGED registry call. The FIB tables are updated from the RIB, removing any stale entries. The RIB starts a failsafe timer during RP switchover, in case the routing protocols fail to notify the RIB of convergence.

The Cisco BGP address family identifier (AFI) model is designed to be modular and scalable, and to support multiple AFI and subsequent address family identifier (SAFI) configurations.

6PE Multipath

Internal and external BGP multipath for IPv6 allows the IPv6 router to load balance between several paths (for example, same neighboring autonomous system [AS] or sub-AS, or the same metric) to reach its destination. The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

When MP-iBGP multipath is enabled on the 6PE router, all labeled paths are installed in the forwarding table with MPLS information (label stack) when MPLS information is available. This functionality enables 6PE to perform load balancing.

How to Implement Multiprotocol BGP for IPv6

When configuring multiprotocol BGP extensions for IPv6, you must create the BGP routing process, configure peering relationships, and customize BGP for your particular network.

**Note**

The following sections describe the configuration tasks for creating an IPv6 multiprotocol BGP routing process and associating peers, peer groups, and networks to the routing process. The following sections do not provide in-depth information on customizing multiprotocol BGP because the protocol functions the same in IPv6 as it does in IPv4. See the “[Related Documents](#)” section for further information on BGP and multiprotocol BGP configuration and command reference information.

The tasks in the following sections explain how to configure multiprotocol BGP extensions for IPv6. Each task in the list is identified as either required or optional:

- [Configuring an IPv6 BGP Routing Process and BGP Router ID, page 4](#) (required)
- [Configuring an IPv6 Multiprotocol BGP Peer, page 5](#) (required)
- [Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address, page 7](#) (optional)
- [Configuring an IPv6 Multiprotocol BGP Peer Group, page 10](#) (optional)
- [Advertising Routes into IPv6 Multiprotocol BGP, page 13](#) (required)
- [Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes, page 14](#) (optional)
- [Redistributing Prefixes into IPv6 Multiprotocol BGP, page 16](#) (optional)
- [Advertising IPv4 Routes Between IPv6 BGP Peers, page 18](#) (optional)
- [Assigning a BGP Administrative Distance, page 20](#) (optional)
- [Generating Translate Updates for IPv6 Multicast BGP, page 21](#) (optional)
- [Configuring the IPv6 BGP Graceful Restart Capability, page 22](#) (optional)
- [Resetting BGP Sessions, page 23](#) (optional)
- [Clearing External BGP Peers, page 24](#) (optional)
- [Clearing IPv6 BGP Route Dampening Information, page 25](#) (optional)
- [Clearing IPv6 BGP Flap Statistics, page 25](#) (optional)
- [Verifying IPv6 Multiprotocol BGP Configuration and Operation, page 26](#) (optional)

Configuring an IPv6 BGP Routing Process and BGP Router ID

This task explains how to configure an IPv6 BGP routing process and an optional BGP router ID for a BGP-speaking router.

Prerequisites

Before configuring the router to run BGP for IPv6, you must globally enable IPv6 routing using the **ipv6 unicast-routing** command. For details on basic IPv6 connectivity tasks, refer to the [Implementing Basic Connectivity for IPv6](#) module.

BGP Router ID for IPv6

BGP uses a router ID to identify BGP-speaking peers. The BGP router ID is 32-bit value that is often represented by an IPv4 address. By default, the Cisco IOS software sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, then the software chooses the highest IPv4 address configured to a physical interface on the router to represent

the BGP router ID. When configuring BGP on a router that is enabled only for IPv6 (the router does not have an IPv4 address), you must manually configure the BGP router ID for the router. The BGP router ID, which is represented as a 32-bit value using an IPv4 address syntax, must be unique to the BGP peers of the router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **bgp router-id** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Configures a BGP routing process, and enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: Router(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. <p>Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as command unless you configure the no bgp default ipv4-unicast command before configuring the neighbor remote-as command.</p>
Step 5	bgp router-id <i>ip-address</i> Example: Router(config-router)# bgp router-id 192.168.99.70	(Optional) Configures a fixed 32-bit router ID as the identifier of the local router running BGP. <p>Note Configuring a router ID using the bgp router-id command resets all active BGP peering sessions.</p>

Configuring an IPv6 Multiprotocol BGP Peer

This task explains how to configure IPv6 multiprotocol BGP between two IPv6 routers (peers).

Restrictions

By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 2001:0DB8:0:CC00::1 remote-as 64600	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. <ul style="list-style-type: none"> • The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
Step 5	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 2001:0DB8:0:CC00::1 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router.

Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address

This task explains how to configure IPv6 multiprotocol BGP between two peers using link-local addresses.

Multiprotocol BGP Peering Using Link-Local Addresses

Configuring IPv6 multiprotocol BGP between two IPv6 routers (peers) using link-local addresses requires that the interface for the neighbor be identified by using the **update-source** command and that a route map be configured to set an IPv6 global next hop.

Restrictions

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*

4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type* *interface-number*
6. **address-family ipv6** [*vrf vrf-name*] [*unicast* | *multicast* | *vpn*]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **route-map** *map-name* {*in* | *out*}
9. **exit**
10. Repeat Step 9.
11. **route-map** *map-tag* [*permit* | *deny*] [*sequence-number*]
12. **match ipv6 address** {*prefix-list prefix-list-name* | *access-list-name*}
13. **set ipv6 next-hop** *ipv6-address* [*link-local-address*] [*peer-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor FE80::XXXX:BFF:FE0E:A471 remote-as 64600	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be a link-local IPv6 address in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type</i> <i>interface-number</i> Example: Router(config-router)# neighbor FE80::XXXX:BFF:FE0E:A471 update-source fastethernet0	Specifies the link-local address over which the peering is to occur. <ul style="list-style-type: none"> If there are multiple connections to the neighbor and you do not specify the neighbor interface by using the <i>interface-type</i> and <i>interface-number</i> arguments in the neighbor update-source command, a TCP connection cannot be established with the neighbor using link-local addresses.

	Command or Action	Purpose
Step 6	address-family ipv6 [vrf vrf-name] [unicast multicast vpnv6] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor FE80::XXXX:BFF:FE0E:A471 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router using the specified link-local addresses.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } route-map <i>map-name</i> { in out } Example: Router(config-router-af)# neighbor FE80::XXXX:BFF:FE0E:A471 route-map nh6 out	Applies a route map to incoming or outgoing routes.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode.
Step 10	Repeat Step 9. Example: Router(config-router)# exit	Exits router configuration mode, and returns the router to global configuration mode.
Step 11	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map nh6 permit 10	Defines a route map and enters route-map configuration mode.

	Command or Action	Purpose
Step 12	<pre>match ipv6 address {prefix-list prefix-list-name access-list-name}</pre> <p>Example: Router(config-route-map)# match ipv6 address prefix-list cisco</p>	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.
Step 13	<pre>set ipv6 next-hop ipv6-address [link-local-address] [peer-address]</pre> <p>Example: Router(config-route-map)# set ipv6 next-hop 2001:0DB8::1</p>	<p>Overrides the next hop advertised to the peer for IPv6 packets that pass a match clause of a route map for policy routing.</p> <ul style="list-style-type: none"> The <i>ipv6-address</i> argument specifies the IPv6 global address of the next hop. It need not be an adjacent router. The <i>link-local-address</i> argument specifies the IPv6 link-local address of the next hop. It must be an adjacent router. <p>Note The route map sets the IPv6 next-hop addresses (global and link-local) in BGP updates. If the route map is not configured, the next-hop address in the BGP updates defaults to the unspecified IPv6 address (::), which is rejected by the peer.</p> <p>If you specify only the global IPv6 next-hop address (the <i>ipv6-address</i> argument) with the set ipv6 next-hop command after specifying the neighbor interface (the <i>interface-type</i> argument) with the neighbor update-source command in Step 5, the link-local address of the interface specified with the <i>interface-type</i> argument is included as the next-hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses.</p>

Troubleshooting Tips

If peering is not established by this task, it may be because of a missing route map **set ipv6 next-hop** command. Use the **debug bgp ipv6 update** command to display debugging information on the updates to help determine the state of the peering.

Configuring an IPv6 Multiprotocol BGP Peer Group

This task explains how to configure an IPv6 peer group to perform multiprotocol BGP routing.

Restrictions

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, peer groups that are defined in router configuration mode using the **neighbor peer-group** command exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, you must activate peer groups using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- Members of a peer group automatically inherit the address prefix configuration of the peer group.
- IPv4 active neighbors cannot exist in the same peer group as active IPv6 neighbors. Create separate peer groups for IPv4 peers and IPv6 peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpnv6**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address*} **send-label**
9. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	router bgp <i>as-number</i>	Enters router configuration mode for the specified BGP routing process.
	Example: Router(config)# router bgp 65000	

	Command or Action	Purpose
Step 4	neighbor <i>peer-group-name</i> peer-group Example: Router(config-router)# neighbor group1 peer-group	Creates a multiprotocol BGP peer group.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 2001:0DB8:0:CC00::1 remote-as 64600	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 6	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Router(config-router)# address-family ipv6 unicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 2001:0DB8:0:CC00::1 activate	Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router. <ul style="list-style-type: none"> To avoid extra configuration steps for each neighbor, use the neighbor activate command with the <i>peer-group-name</i> argument as an alternative in this step.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> } send-label Example: Router(config-router-af)# neighbor 192.168.99.70 send-label	Advertises the capability of the router to send MPLS labels with BGP routes. <ul style="list-style-type: none"> In IPv6 address family configuration mode, this command enables binding and advertisement of aggregate labels when advertising IPv6 prefixes in BGP.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> } peer-group <i>peer-group-name</i> Example: Router(config-router-af)# neighbor 2001:0DB8:0:CC00::1 peer-group group1	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 10	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. <ul style="list-style-type: none"> Repeat this step to exit router configuration mode and return the router to global configuration mode.

What to Do Next

Refer to the section “Configure BGP Peer Groups” of the “Configuring BGP” chapter in *Cisco IOS IP Configuration Guide*, Release 12.4, for more information on assigning options to peer groups and making a BGP or multiprotocol BGP neighbor a member of a peer group.

Advertising Routes into IPv6 Multiprotocol BGP

This task explains how to advertise (inject) a prefix into IPv6 multiprotocol BGP.

Restrictions

By default, networks that are defined in router configuration mode using the **network** command are injected into the IPv4 unicast database. To inject a network into another database, such as the IPv6 BGP database, you must define the network using the **network** command in address family configuration mode for the other database, as shown for the IPv6 BGP database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	router bgp <i>as-number</i>	Enters router configuration mode for the specified BGP routing process.
	Example: Router(config)# router bgp 65000	

	Command or Action	Purpose
Step 4	address-family ipv6 [vrf <i>vrf-name</i>] [unicast multicast vpn <i>v6</i>] Example: Router(config-router)# address-family ipv6 unicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Router(config-router-af)# network 2001:0DB8::/24	Advertises (injects) the specified prefix into the IPv6 BGP database. (The routes must first be found in the IPv6 unicast routing table.) <ul style="list-style-type: none"> Specifically, the prefix is injected into the database for the address family specified in the previous step. Routes are tagged from the specified prefix as “local origin.” The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 6	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. <ul style="list-style-type: none"> Repeat this step to exit router configuration mode and return the router to global configuration mode.

Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes

This task explains how to configure a route map for IPv6 multiprotocol BGP prefixes.

Restrictions

- By default, neighbors that are defined using the **neighbor remote-as** command in router configuration mode exchange only IPv4 unicast address prefixes. To exchange other address prefix types, such as IPv6 prefixes, neighbors must also be activated using the **neighbor activate** command in address family configuration mode for the other prefix types, as shown for IPv6 prefixes.
- By default, route maps that are applied in router configuration mode using the **neighbor route-map** command are applied to only IPv4 unicast address prefixes. Route maps for other address families must be applied in address family configuration mode using the **neighbor route-map** command, as shown for the IPv6 address family. The route maps are applied either as the inbound or outbound

routing policy for neighbors under the specified address family. Configuring separate route maps under each address family type simplifies managing complicated or different policies for each address family.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **route-map** *map-name* {**in** | **out**}
8. **exit**
9. Repeat Step 8.
10. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
11. **match ipv6 address** {**prefix-list** *prefix-list-name* | *access-list-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 2001:0DB8:0:cc00::1 remote-as 64600	Adds the link-local IPv6 address of the neighbor in the specified remote autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be a link-local IPv6 address in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.

	Command or Action	Purpose
Step 5	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 2001:0DB8:0:cc00::1 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router using the specified link-local addresses.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } route-map <i>map-name</i> { in out } Example: Router(config-router-af)# neighbor 2001:0DB8:0:cc00::1 route-map rtp in	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.
Step 8	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode.
Step 9	Repeat Step 8. Example: Router(config-router)# exit	Exits router configuration mode, and returns the router to global configuration mode.
Step 10	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map rtp permit 10	Defines a route map and enters route-map configuration mode. <ul style="list-style-type: none"> Follow this step with a match command.
Step 11	match ipv6 address { <i>prefix-list</i> <i>access-list-name</i> } Example: Router(config-route-map)# match ipv6 address prefix-list cisco	Distributes any routes that have a destination IPv6 network number address permitted by a prefix list, or performs policy routing on packets.

Redistributing Prefixes into IPv6 Multiprotocol BGP

This task explains how to redistribute (inject) prefixes from another routing protocol into IPv6 multiprotocol BGP.

Redistribution for IPv6

Redistribution is the process of injecting prefixes from one routing protocol into another routing protocol. This task explains how to inject prefixes from a routing protocol into IPv6 multiprotocol BGP. Specifically, prefixes that are redistributed into IPv6 multiprotocol BGP using the **redistribute** router configuration command are injected into the IPv6 unicast database.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpnvp6**]
5. **redistribute bgp** [*process-id*] [[**metric** *metric-value*] [**route-map** *map-name*]] [*source-protocol-options*]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpnvp6] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> • The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. • The multicast keyword specifies IPv6 multicast address prefixes.

	Command or Action	Purpose
Step 5	redistribute bgp [<i>process-id</i>] [[metric <i>metric-value</i>] [route-map <i>map-name</i>]] [<i>source-protocol-options</i>] Example: Router(config-router-af)# redistribute bgp 64500 metric 5 metric-type external	Redistributes IPv6 routes from one routing domain into another routing domain.
Step 6	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. <ul style="list-style-type: none"> Repeat this step to exit router configuration mode and return the router to global configuration mode.

Advertising IPv4 Routes Between IPv6 BGP Peers

This task explains how to advertise IPv4 routes between IPv6 peers. If an IPv6 network is connecting two separate IPv4 networks, it is possible to use IPv6 to advertise the IPv4 routes. Configure the peering using the IPv6 addresses within the IPv4 address family. Set the next hop with a static route or with an inbound route map because the advertised next hop will usually be unreachable. Advertising IPv6 routes between two IPv4 peers is also possible using the same model.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** *ipv6-address* **peer-group** *peer-group-name*
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **route-map** *map-name* {**in** | **out**}
9. **exit**
10. Repeat Step 11.
11. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
12. **set ip next-hop** *ip-address* [... *ip-address*] [*peer-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp as-number Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	neighbor peer-group-name peer-group Example: Router(config-router)# neighbor 6peers peer-group	Creates a multiprotocol BGP peer group.
Step 5	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 6peers remote-as 65002	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multiprotocol BGP neighbor table of the local router. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 6	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor ipv6-address peer-group peer-group-name Example: Router(config-router-af)# neighbor 2001:0DB8:yyyy::2 peer-group 6peers	Assigns the IPv6 address of a BGP neighbor to a peer group.
Step 8	neighbor {ip-address peer-group-name ipv6-address} route-map map-name {in out} Example: Router(config-router-af)# neighbor 6peers route-map rmap out	Applies a route map to incoming or outgoing routes. <ul style="list-style-type: none"> Changes to the route map will not take effect for existing peers until the peering is reset or a soft reset is performed. Using the clear bgp ipv6 command with the soft and in keywords will perform a soft reset.

	Command or Action	Purpose
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode.
Step 10	Repeat Step 11. Example: Router(config-router)# exit	Exits router configuration mode, and returns the router to global configuration mode.
Step 11	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map rmap permit 10	Defines a route map and enters route-map configuration mode.
Step 12	set ip next-hop ip-address [... <i>ip-address</i>] [<i>peer-address</i>] Example: Router(config-route-map)# set ip next-hop 10.21.8.10	Overrides the next hop advertised to the peer for IPv4 packets.

Assigning a BGP Administrative Distance

This task explains how to specify an administrative distance for multicast BGP routes to be used in RPF lookups for comparison with unicast routes.



Caution

Changing the administrative distance of BGP internal routes is not recommended. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **distance bgp** *external-distance internal-distance local-distance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [vrf <i>vrf-name</i>] [unicast multicast vpn <i>vpn6</i>] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	distance bgp <i>external-distance</i> <i>internal-distance</i> <i>local-distance</i> Example: Router(config-router-af)# distance bgp 10 50 100	Configures the administrative distance for BGP routes.

Generating Translate Updates for IPv6 Multicast BGP

This task explains how to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates received from a peer.

The MBGP translate-update feature generally is used in an MBGP-capable router that peers with a customer site that has only a BGP-capable router; the customer site has not or cannot upgrade its router to an MBGP-capable image. Because the customer site cannot originate MBGP advertisements, the router with which it peers will translate the BGP prefixes into MBGP prefixes, which are used for multicast-source Reverse Path Forwarding (RPF) lookup.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*

4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]
5. **neighbor ipv6-address translate-update ipv6 multicast** [**unicast**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast vpn6] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	neighbor ipv6-address translate-update ipv6 multicast [unicast] Example: Router(config-router-af)# neighbor 7000::2 translate-update ipv6 multicast	Generates multiprotocol IPv6 BGP updates that correspond to unicast IPv6 updates received from a peer.

Configuring the IPv6 BGP Graceful Restart Capability

This task explains how to configure the IPv6 BGP graceful restart capability, therefore enabling NSF functionality.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [*vrf vrf-name*] [**unicast** | **multicast** | **vpn6**]

5. `bgp graceful-restart` [`restart-time seconds` | `stalepath-time seconds`] [`all`]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [vrf <i>vrf-name</i>] [unicast multicast vpnv6] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family.
Step 5	bgp graceful-restart [restart-time <i>seconds</i> stalepath-time <i>seconds</i>] [all] Example: Router(config-router)# bgp graceful-restart	Enables the BGP graceful restart capability.

Resetting BGP Sessions

This task explains how to reset IPv6 BGP sessions.

SUMMARY STEPS

- enable**
- clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | *peer-group-name*} [**soft**] [**in** | **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} {* autonomous-system-number ip-address ipv6-address peer-group-name} [soft] [in out] Example: Router# clear bgp ipv6 unicast peer-group marketing soft out	Resets IPv6 BGP sessions.

Clearing External BGP Peers

This task explains how to clear external BGP peers and members of an IPv6 BGP peer group.

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6 {unicast | multicast} external [soft] [in | out]**
3. **clear bgp ipv6 {unicast | multicast} peer-group [name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} external [soft] [in out] Example: Router# clear bgp ipv6 unicast external soft in	Clears external IPv6 BGP peers.
Step 3	clear bgp ipv6 {unicast multicast} peer-group [name] Example: Router# clear bgp ipv6 unicast peer-group	Clears all members of an IPv6 BGP peer group.

Clearing IPv6 BGP Route Dampening Information

This task explains how to clear IPv6 BGP route dampening information and how to unsuppress suppressed routes.

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} dampening [ipv6-prefix/prefix-length]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} dampening [ipv6-prefix/prefix-length] Example: Router# clear bgp ipv6 unicast dampening 2001:0DB8::/64	Clears IPv6 BGP route dampening information and unsuppress the suppressed routes.

Clearing IPv6 BGP Flap Statistics

This task explains how to clear IPv6 BGP flap statistics.

SUMMARY STEPS

1. `enable`
2. `clear bgp ipv6 {unicast | multicast} flap-statistics [ipv6-prefix/prefix-length | regexp regexp | filter-list list]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} flap-statistics [ipv6-prefix/prefix-length regexp regexp filter-list list] Example: Router# clear bgp ipv6 unicast flap-statistics filter-list 3	Clears IPv6 BGP flap statistics.

Verifying IPv6 Multiprotocol BGP Configuration and Operation

This task explains how to display information to verify the configuration and operation of IPv6 multiprotocol BGP.

SUMMARY STEPS

1. **show bgp ipv6 {unicast | multicast} [ipv6-prefix/prefix-length] [longer-prefixes] [labels]**
2. **show bgp ipv6 {unicast | multicast} summary**
3. **show bgp ipv6 {unicast | multicast} dampening dampened-paths**
4. **enable**
5. **debug bgp ipv6 {unicast | multicast} dampening [prefix-list prefix-list-name]**
6. **debug bgp ipv6 {unicast | multicast} updates [ipv6-address] [prefix-list prefix-list-name] [in | out]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	show bgp ipv6 {unicast multicast} [ipv6-prefix/prefix-length] [longer-prefixes] [labels] Example: Router> show bgp ipv6 unicast	(Optional) Displays entries in the IPv6 BGP routing table.
Step 2	show bgp ipv6 {unicast multicast} summary Example: Router> show bgp ipv6 unicast summary	(Optional) Displays the status of all IPv6 BGP connections.

	Command or Action	Purpose
Step 3	show bgp ipv6 {unicast multicast} dampening dampened-paths Example: Router> show bgp ipv6 unicast dampening dampened-paths	(Optional) Displays IPv6 BGP dampened routes.
Step 4	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 5	debug bgp ipv6 {unicast multicast} dampening [prefix-list prefix-list-name] Example: Router# debug bgp ipv6 unicast dampening	(Optional) Displays debugging messages for IPv6 BGP dampening packets. <ul style="list-style-type: none"> If no prefix list is specified, debugging messages for all IPv6 BGP dampening packets are displayed.
Step 6	debug bgp ipv6 {unicast multicast} updates [ipv6-address] [prefix-list prefix-list-name] [in out] Example: Router# debug bgp ipv6 unicast updates	(Optional) Displays debugging messages for IPv6 BGP update packets. <ul style="list-style-type: none"> If an <i>ipv6-address</i> argument is specified, debugging messages for IPv6 BGP updates to the specified neighbor are displayed. Use the in keyword to display debugging messages for inbound updates only. Use the out keyword to display debugging messages for outbound updates only.

Examples

This section provides the following output examples:

- [Sample Output for the show bgp ipv6 Command](#)
- [Sample Output for the show bgp ipv6 summary Command](#)
- [Sample Output for the show bgp ipv6 dampened-paths Command](#)
- [Sample Output for the debug bgp ipv6 dampening Command](#)
- [Sample Output for the debug bgp ipv6 updates Command](#)

Sample Output for the show bgp ipv6 Command

In the following example, entries in the IPv6 BGP routing table are displayed using the **show bgp ipv6** command:

```
Router> show bgp ipv6 unicast
```

```
BGP table version is 12612, local router ID is 192.168.99.70
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```

      Network          Next Hop           Metric LocPrf Weight Path
*>
      2001:0DB8:E:C::2              0 3748 4697 1752 i
```

```

*                2001:0DB8:0:CC00::1
* 2001:618:3::/48 2001:0DB8:E:4::2      1
*>                2001:0DB8:0:CC00::1
*                2001:0DB8:0:F004::1
*> 2001:620::/35  2001:0DB8:0:F004::1
*                2001:0DB8:E:9::2
*                2001:0DB8::A
*                2001:0DB8:20:1::11
*                2001:0DB8:E:4::2      1
*                2001:0DB8:E:B::2
*                2001:0DB8:E:C::2
0 1849 1273 1752 i
0 4554 1849 65002 i
0 1849 65002 i
0 3320 1275 559 i
0 1251 1930 559 i
0 3462 10566 1930 559 i
0 293 1275 559 i
0 4554 1849 1273 559 i
0 237 3748 1275 559 i
0 3748 1275 559 i

```

Sample Output for the show bgp ipv6 summary Command

In the following example, the status of all IPv6 BGP connections is displayed using the **show bgp ipv6 summary** command with the **unicast** keyword:

```
Router# show bgp ipv6 unicast summary
```

```
BGP router identifier 172.30.4.4, local AS number 200
BGP table version is 1, main routing table version 1
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
2001:0DB8:101::2	4	200	6869	6882	0	0	0	06:25:24	Active

Sample Output for the show bgp ipv6 dampened-paths Command

In the following example, IPv6 BGP dampened routes are displayed using the **show bgp ipv6 dampened-paths** command with the **unicast** keyword:

```
Router# show bgp ipv6 unicast dampening dampened-paths
```

```
BGP table version is 12610, local router ID is 192.168.7.225
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	From	Reuse	Path
*d 3FFE:1000::/24	3FFE:C00:E:B::2	00:00:10	237 2839 5609 i
*d 2001:228::/35	3FFE:C00:E:B::2	00:23:30	237 2839 5609 2713 i

Sample Output for the debug bgp ipv6 dampening Command

In the following example, debugging messages for IPv6 BGP dampening packets are displayed using the **debug bgp ipv6 dampening** command with the **unicast** keyword:



Note

By default, the system sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within configuration mode. Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server. For complete information on **debug** commands and redirecting debugging output, refer to *Cisco IOS Debug Command Reference*, Release 12.4.

```
Router# debug bgp ipv6 unicast dampening
```

```

00:13:28:BGP(1):charge penalty for 2001:0DB8:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2001:0DB8:0:1:1::/80 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:13:28:BGP(1):charge penalty for 2001:0DB8:0:5::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:13:28:BGP(1):flapped 1 times since 00:00:00. New penalty is 1000
00:16:03:BGP(1):charge penalty for 2001:0DB8:0:1::/64 path 2 1 with halflife-time 15
reuse/suppress 750/2000
00:16:03:BGP(1):flapped 2 times since 00:02:35. New penalty is 1892
00:18:28:BGP(1):suppress 2001:0DB8:0:1:1::/80 path 2 1 for 00:27:30 (penalty 2671)
00:18:28:halflife-time 15, reuse/suppress 750/2000
00:18:28:BGP(1):suppress 2001:0DB8:0:1::/64 path 2 1 for 00:27:20 (penalty 2664)
00:18:28:halflife-time 15, reuse/suppress 750/2000

```

Sample Output for the debug bgp ipv6 updates Command

In the following example, debugging messages for IPv6 BGP update packets are displayed using the **debug bgp ipv6 updates** command with the **unicast** keyword:

```
Router# debug bgp ipv6 unicast updates
```

```

14:04:17:BGP(1):2001:0DB8:0:2::2 computing updates, afi 1, neighbor version 0, table
version 1, starting at ::
14:04:17:BGP(1):2001:0DB8:0:2::2 update run completed, afi 1, ran for 0ms, neighbor
version 0, start version 1, throttled to 1
14:04:19:BGP(1):sourced route for 2001:0DB8:0:2::1/64 path #0 changed (weight 32768)
14:04:19:BGP(1):2001:0DB8:0:2::1/64 route sourced locally
14:04:19:BGP(1):2001:0DB8:0:2:1::/80 route sourced locally
14:04:19:BGP(1):2001:0DB8:0:3::2/64 route sourced locally
14:04:19:BGP(1):2001:0DB8:0:4::2/64 route sourced locally
14:04:22:BGP(1):2001:0DB8:0:2::2 computing updates, afi 1, neighbor version 1, table
version 6, starting at ::
14:04:22:BGP(1):2001:0DB8:0:2::2 send UPDATE (format) 2001:0DB8:0:2::1/64, next
2001:0DB8:0:2::1, metric 0, path
14:04:22:BGP(1):2001:0DB8:0:2::2 send UPDATE (format) 2001:0DB8:0:2:1::/80, next
2001:0DB8:0:2::1, metric 0, path
14:04:22:BGP(1):2001:0DB8:0:2::2 send UPDATE (prepend, chgflags:0x208)
2001:0DB8:0:3::2/64, next 2001:0DB8:0:2::1, metric 0, path
14:04:22:BGP(1):2001:0DB8:0:2::2 send UPDATE (prepend, chgflags:0x208)
2001:0DB8:0:4::2/64, next 2001:0DB8:0:2::1, metric 0, path

```

Configuration Examples for Multiprotocol BGP for IPv6

This section provides the following configuration examples:

- [Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer: Example, page 30](#)
- [Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address: Example, page 30](#)
- [Configuring an IPv6 Multiprotocol BGP Peer Group: Example, page 30](#)
- [Advertising Routes into IPv6 Multiprotocol BGP: Example, page 31](#)
- [Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes: Example, page 31](#)
- [Redistributing Prefixes into IPv6 Multiprotocol BGP: Example, page 31](#)
- [Advertising IPv4 Routes Between IPv6 Peers: Example, page 31](#)

Configuring a BGP Process, BGP Router ID, and IPv6 Multiprotocol BGP Peer: Example

The following example enables IPv6 globally, configures a BGP process, and establishes a BGP router ID. Also, the IPv6 multiprotocol BGP peer 2001:0DB8:0:CC00:: is configured and activated.

```
ipv6 unicast-routing
!
router bgp 65000
no bgp default ipv4-unicast
bgp router-id 192.168.99.70
neighbor 2001:0DB8:0:CC00::1 remote-as 64600

address-family ipv6 unicast
neighbor 2001:0DB8:0:CC00::1 activate
```

Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address: Example

The following example configures the IPv6 multiprotocol BGP peer FE80::XXXX:BFF:FE0E:A471 over Fast Ethernet interface 0 and sets the route map named nh6 to include the IPv6 next-hop global address of Fast Ethernet interface 0 in BGP updates. The IPv6 next-hop link-local address can be set by the nh6 route map (not shown in the following example) or from the interface specified by the **neighbor update-source** command (as shown in the following example).

```
router bgp 65000
neighbor FE80::XXXX:BFF:FE0E:A471 remote-as 64600
neighbor FE80::XXXX:BFF:FE0E:A471 update-source fastethernet 0

address-family ipv6
neighbor FE80::XXXX:BFF:FE0E:A471 activate
neighbor FE80::XXXX:BFF:FE0E:A471 route-map nh6 out

route-map nh6 permit 10
match ipv6 address prefix-list cisco
set ipv6 next-hop 2001:0DB8:5y6::1

ipv6 prefix-list cisco permit 2001:0DB8:2Fy2::/48 le 128
ipv6 prefix-list cisco deny ::/0
```



Note

If you specify only the global IPv6 next-hop address (the *ipv6-address* argument) with the **set ipv6 next-hop** command after specifying the neighbor interface (the *interface-type* argument) with the **neighbor update-source** command, the link-local address of the interface specified with the *interface-type* argument is included as the next hop in the BGP updates. Therefore, only one route map that sets the global IPv6 next-hop address in BGP updates is required for multiple BGP peers that use link-local addresses.

Configuring an IPv6 Multiprotocol BGP Peer Group: Example

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
```

```
neighbor 2001:0DB8:0:CC00::1 remote-as 64600

address-family ipv6 unicast
neighbor group1 activate
neighbor 2001:0DB8:0:CC00::1 peer-group group1
```

Advertising Routes into IPv6 Multiprotocol BGP: Example

The following example injects the IPv6 network 2001:0DB8::/24 into the IPv6 unicast database of the local router. (BGP checks that a route for the network exists in the IPv6 unicast database of the local router before advertising the network.)

```
router bgp 65000
no bgp default ipv4-unicast

address-family ipv6 unicast
network 2001:0DB8::/24
```

Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes: Example

The following example configures the route map named rtp to permit IPv6 unicast routes from network 2001:0DB8::/24 if they match the prefix list named cisco:

```
router bgp 64900
no bgp default ipv4-unicast
neighbor 2001:0DB8:0:CC00::1 remote-as 64700

address-family ipv6 unicast
neighbor 2001:0DB8:0:CC00::1 activate
neighbor 2001:0DB8:0:CC00::1 route-map rtp in

ipv6 prefix-list cisco seq 10 permit 2001:0DB8::/24

route-map rtp permit 10
match ipv6 address prefix-list cisco
```

Redistributing Prefixes into IPv6 Multiprotocol BGP: Example

The following example redistributes RIP routes into the IPv6 unicast database of the local router:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 unicast
redistribute rip
```

Advertising IPv4 Routes Between IPv6 Peers: Example

The following example advertises IPv4 routes between IPv6 peers when the IPv6 network is connecting two separate IPv4 networks. Peering is configured using IPv6 addresses in the IPv4 address family configuration mode. The inbound route map named rmap sets the next hop because the advertised next hop is likely to be unreachable.

```
router bgp 65000
!
neighbor 6peers peer-group
```

```
neighbor 2001:0DB8:yyyy::2 remote-as 65002
address-family ipv4
neighbor 6peers activate
neighbor 6peers soft-reconfiguration inbound
neighbor 2001:0DB8:yyyy::2 peer-group 6peers
neighbor 2001:0DB8:yyyy::2 route-map rmap in
!
route-map rmap permit 10
set ip next-hop 10.21.8.10
```

Where to Go Next

If you want to implement more IPv6 routing protocols, refer to the [Implementing RIP for IPv6](#) or the [Implementing IS-IS for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing Multiprotocol BGP for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv4 BGP configuration tasks	“BGP Features Roadmap,” Cisco IOS IP Routing Protocols Configuration Guide
Multiprotocol BGP configuration tasks	“BGP Features Roadmap,” Cisco IOS IP Routing Protocols Configuration Guide
BGP and multiprotocol BGP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	“BGP Commands,” Cisco IOS IP Routing Protocols Command Reference
Cisco Nonstop Forwarding	“Cisco Nonstop Forwarding,” Cisco IOS High Availability Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
RFC 2858	Multiprotocol Extensions for BGP-4
RFC 4007	IPv6 Scoped Address Architecture
RFC 4364	BGP MPLS/IP Virtual Private Networks (VPNs)
RFC 4382	MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base

RFCs	Title
RFC 4659	<i>BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN</i>
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **address-family ipv4 (BGP)**
- **address-family ipv6**
- **bgp graceful-restart**
- **clear bgp ipv6**
- **clear bgp ipv6 dampening**
- **clear bgp ipv6 external**
- **clear bgp ipv6 flap-statistics**
- **clear bgp ipv6 peer-group**
- **debug bgp ipv6 dampening**
- **debug bgp ipv6 updates**
- **debug ipv6 routing**
- **distance bgp (IPv6)**
- **match ipv6 address**

- **maximum-paths (IPv6)**
- **neighbor activate**
- **neighbor peer-group (assigning members)**
- **neighbor peer-group (creating)**
- **neighbor remote-as**
- **neighbor route-map**
- **neighbor send-label**
- **neighbor translate-update**
- **neighbor update-source**
- **network (BGP and multiprotocol BGP)**
- **redistribute (IPv6)**
- **route-map**
- **router bgp**
- **set ipv6 next-hop (BGP)**
- **show bgp ipv6**
- **show bgp ipv6 community**
- **show bgp ipv6 community-list**
- **show bgp ipv6 dampened-paths**
- **show bgp ipv6 filter-list**
- **show bgp ipv6 flap-statistics**
- **show bgp ipv6 inconsistent-as**
- **show bgp ipv6 labels**
- **show bgp ipv6 neighbors**
- **show bgp ipv6 paths**
- **show bgp ipv6 peer-group**
- **show bgp ipv6 prefix-list**
- **show bgp ipv6 quote-regexp**
- **show bgp ipv6 regexp**
- **show bgp ipv6 route-map**
- **show bgp ipv6 summary**
- **synchronization (IPv6)**

Feature Information for Implementing Multiprotocol BGP for IPv6

Table 6 lists the release history for this feature.

For information on a feature in this technology that is not documented here, see “Start Here: Cisco IOS Software Release Specifies for IPv6 Features.”

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 6 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 6 Feature Information for Implementing Multiprotocol BGP for IPv6

Feature Name	Releases	Feature Information
IPv6 routing: multiprotocol BGP extensions for IPv6	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Multiprotocol BGP extensions for IPv6 supports the same features and functionality as IPv4 BGP. The following sections provide information about this feature: <ul style="list-style-type: none"> • Multiprotocol BGP Extensions for IPv6, page 2 • How to Implement Multiprotocol BGP for IPv6, page 3
IPv6 routing: multiprotocol BGP link-local address peering	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(4)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IPv6 supports multiprotocol BGP link-local address peering. The following sections provide information about this feature: <ul style="list-style-type: none"> • Configuring an IPv6 Multiprotocol BGP Peer Using a Link-Local Address, page 7 • Multiprotocol BGP Peering Using Link-Local Addresses, page 7

Table 6 **Feature Information for Implementing Multiprotocol BGP for IPv6**

Feature Name	Releases	Feature Information
Advertising routes into IPv6 multiprotocol BGP	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	Users advertise (inject) a prefix into IPv6 multiprotocol BGP. The following sections provide information about this feature: <ul style="list-style-type: none"> • Advertising Routes into IPv6 Multiprotocol BGP, page 13 • Advertising Routes into IPv6 Multiprotocol BGP: Example, page 31
Configuring route maps for IPv6 multiprotocol BGP prefixes	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	Users can configure route maps for IPv6 multiprotocol BGP prefixes. The following sections provide information about this feature: <ul style="list-style-type: none"> • Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes, page 14 • Configuring a Route Map for IPv6 Multiprotocol BGP Prefixes: Example, page 31
Redistributing prefixes into IPv6 multiprotocol BGP	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	Users can redistribute (inject) prefixes from another routing protocol into IPv6 multiprotocol BGP. The following sections provide information about this feature: <ul style="list-style-type: none"> • Redistributing Prefixes into IPv6 Multiprotocol BGP, page 16 • Redistributing Prefixes into IPv6 Multiprotocol BGP: Example, page 31
IPv6 multicast address family support for multiprotocol BGP	12.0(26)S 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(4)T 12.4 12.4(2)T	The multiprotocol BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. The following sections provide information about this feature: <ul style="list-style-type: none"> • Multiprotocol BGP for the IPv6 Multicast Address Family, page 2 • How to Implement Multiprotocol BGP for IPv6, page 3

Table 6 **Feature Information for Implementing Multiprotocol BGP for IPv6**

Feature Name	Releases	Feature Information
6PE multipath	12.2(25)S 12.2(28)SB 12.2(33)SRA 12.4(6)T	The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route. The following sections provide information about this feature: <ul style="list-style-type: none"> • 6PE Multipath, page 3
IPv6: NSF and graceful restart for MP-BGP IPv6 address family	Cisco IOS XE Release 2.1	IPv6 BGP supports Cisco Nonstop Forwarding and graceful restart. The following sections provide information about this feature: <ul style="list-style-type: none"> • Multiprotocol BGP for the IPv6 Multicast Address Family, page 2 • Configuring the IPv6 BGP Graceful Restart Capability, page 22

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing DHCP for IPv6

First Published: June 26, 2006

Last Updated: February 27, 2009

The “Implementing DHCP for IPv6” module describes how to configure Dynamic Host Configuration Protocol (DHCP) for IPv6 prefix delegation on your networking devices.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing DHCP for IPv6”](#) section on page 40.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Introduction](#), page 2
- [Restrictions for Implementing DHCP for IPv6](#), page 2
- [Information About Implementing DHCP for IPv6](#), page 2
- [How to Implement DHCP for IPv6](#), page 8
- [Configuration Examples for Implementing DHCP for IPv6](#), page 33
- [Additional References](#), page 36
- [Command Reference](#), page 37
- [Feature Information for Implementing DHCP for IPv6](#), page 40



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing DHCP for IPv6

Restrictions for Implementing DHCP for IPv6

- Cisco IOS Release 12.0S provides IPv6 support on Cisco 12000 series Internet routers and Cisco 10720 Internet routers only.
- The DHCPv6 Remote-ID for Ethernet interfaces feature works only for Ethernet interfaces in Cisco IOS Release 12.2(33)SRC.

The DHCPv6 implementation in Cisco IOS Release 12.3(4)T, Cisco IOS Release 12.0(32)S, and Cisco IOS 12.2(33)SRC support only stateless address assignment.

Information About Implementing DHCP for IPv6

-

DHCPv6 Prefix Delegation

- Stateful—Address assignment is centrally managed and clients must obtain configuration information not available through protocols such as address autoconfiguration and neighbor discovery.
- Stateless—Stateless configuration parameters do not require a server to maintain any dynamic state for individual clients, such as Domain Name System (DNS) server addresses and domain search list options.

Extensions to DHCPv6 also enable prefix delegation, through which an Internet service provider (ISP) can automate the process of assigning prefixes to a customer for use within the customer's network. Prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE), using the DHCPv6 prefix delegation option. Once the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer's network.

Configuring Nodes Without Prefix Delegation

Client and Server Identification

addresses for both the client and server identifier. The device uses the MAC address from the lowest-numbered interface to form the DUID. The network interface is assumed to be permanently attached to the device.

When an IPv6 DHCP client requests two prefixes with the same DUID but different IAIDs on two different interfaces, these prefixes are considered to be for two different clients, and interface information is maintained for both.

Rapid Commit

request, reply). By default, the four-message exchange is used. When the rapid-commit option is enabled by both client and server, the two-message exchange is used.

DHCPv6 Client, Server, and Relay Functions

-
-
-

Client Function

be referred to from other applications; for example, the general prefix pools can be used to number router downstream interfaces.

Server Selection

IAPD and IAID

Server Function

pools, which are stored in NVRAM. A configuration pool can be associated with a particular DHCPv6 server on an interface when it is started. Prefixes to be delegated to clients may be specified either as a list of preassigned prefixes for a particular client or as IPv6 local prefix pools that are also stored in NVRAM. The list of manually configured prefixes or IPv6 local prefix pools can be referenced and used by DHCPv6 configuration pools.

The DHCPv6 server maintains an automatic binding table in memory to track the assignment of some configuration parameters, such as prefixes between the server and its clients. The automatic bindings can be stored permanently in the database agent, which can be, for example, a remote TFTP server or local NVRAM file system.

Configuration Information Pool

through the command-line interface (CLI).

Each configuration pool can contain the following configuration parameters and operational information:

- Prefix delegation information, which could include:
 -
 -

DHCP for IPv6 Address Assignment

Prefix Assignment

Implementing ADSL and Deploying Dial Access for IPv6

Automatic Binding

-
-
-
-
-
-
-

prefixes' valid lifetimes have expired, or administrators run the **clear ipv6 dhcp binding**

Binding Database

-
-
-
-
-
-

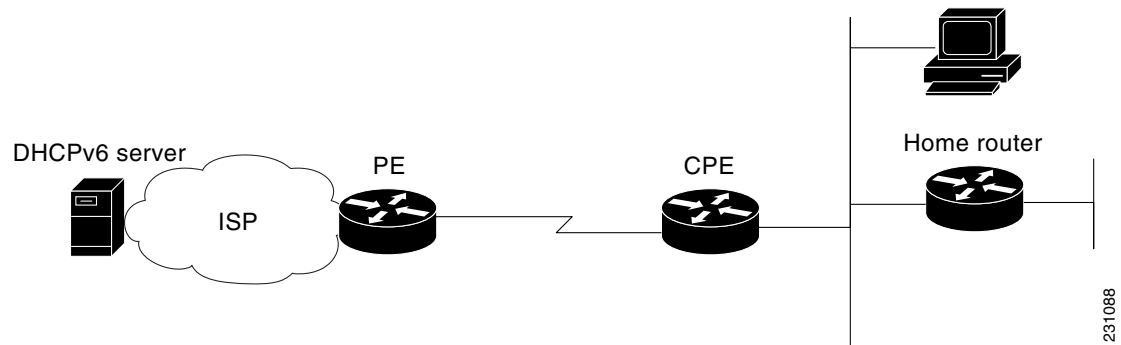
of the file, after the text records, the text string “*end*” is stored to detect file truncation.

The permanent storage to which the binding database is saved is called the database agent. Database agents include FTP and TFTP servers, RCP, flash file system, and NVRAM.

Hierarchical DHCPv6 for stateless configuration parameters allows a stateless or stateful DHCPv6 client to export configuration parameters (DHCPv6 options) to a local DHCPv6 server pool. The local DHCPv6 server can then provide the imported configuration parameters to other DHCPv6 clients.

[Figure 1](#) shows a typical broadband deployment.

Figure 1 **Broadband Topology**



The CPE interface toward the PE can be a stateless or stateful DHCPv6 client. In either case, the ISP-side DHCPv6 server may provide configuration parameters such as DNS server addresses, domain names, and Simple Network Time Protocol (SNTP) servers to the DHCP client on the CPE. These information can be specific to an ISP and may change.

In addition to being a DHCPv6 client (for example, toward the ISP), the CPE may act as a DHCPv6 server to the home network. For example, Neighbor Discovery followed by stateless or stateful DHCPv6 can occur on the link between CPE and the home devices (for example, the home router or PC). In some cases, the information to be provided to the home network is the same information obtained from the ISP-side DHCPv6 server. Because this information can be dynamically changed, it cannot be hard-configured in the CPE's configuration. Therefore, the DHCPv6 component on the CPE allows automatic importing of configuration parameters from the DHCPv6 client to the DHCPv6 server pool.

DHCPv6 provides support of the options for IPv6 on the server described in the following sections:

[Information Refresh Server Option, page 7](#)

[NIS- and NIS+-Related Server Options, page 7](#)

[SIP Server Options, page 7](#)

[SNTP Server Option, page 7](#)

Information Refresh Server Option

NIS- and NIS+-Related Server Options

SIP Server Options

SNTP Server Option

client for synchronization. The clients use these SNTP servers to synchronize their system time to that of the standard time servers. The server may list the SNTP servers in decreasing order of preference, but clients must treat the list of SNTP servers as an ordered list.

DHCP Relay Agent

DHCPv6 Relay Agent Notification for Prefix Delegation

to other nodes. The static routes will be removed when an DHCP_DECLINE message is sent by the client.

DHCPv6 Relay Options: Remote-ID for Ethernet Interfaces

DHCPv6 Relay Options: Reload Persistent Interface-ID

How to Implement DHCP for IPv6

- [Configuring the DHCPv6 Client Function, page 9](#) (required)
- [Configuring the DHCPv6 Client Function, page 11](#) (required)
- [Configuring the DHCPv6 Relay Agent, page 12](#) (required)
- [Configuring DHCP for IPv6 Address Assignment, page 13](#) (required)
- [Verifying DHCPv6 Configuration and Operation, page 30](#) (optional)
- [Configuring the Stateless DHCPv6 Function, page 16](#) (required)
- [Configuring the DHCPv6 Server Options, page 19](#) (required)

- Defining a General Prefix with the DHCPv6 Prefix Delegation Client Function, page 27 (required)
- Restarting the DHCPv6 Client on an Interface, page 28 (optional)
- Deleting Automatic Client Bindings from the DHCPv6 Binding Table, page 28 (required)
- Troubleshooting DHCPv6, page 29 (optional)
- Verifying DHCPv6 Configuration and Operation, page 30 (optional)

Configuring the DHCPv6 Server Function

-
-

Configuring DHCPv6 Configuration Pool

SUMMARY STEPS

-
- configure terminal**
- ipv6 dhcp pool**
- domain-name**
- dns-server** *ipv6-address*
prefix-delegation *ipv6-prefix/prefix-length client-DUID* [*iaid*] []
[{ | }]
-
-
-
- number*
- poolname* *value* **allow-hint**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. Enter your password if prompted.
	Router> enable	
	configure terminal	
	Router# configure terminal	

Step 3

ipv6 dhcp pool *poolname*

Router(config)# **ipv6 dhcp pool** pool1

domain-name *domain*

Router(config-dhcp)# **domain-name** example.com

dns-server *ipv6-address*

Router(config-dhcp)# **dns-server**
2001:0DB8:3000:3000::42

prefix-delegation *ipv6-prefix/prefix-length*
client-DUID [*iaid*] []

Router(config-dhcp)# **prefix-delegation**
2001:0DB8:1263::/48 0005000400F1A4D070D03

{ | [] }

Example:

exit

interface *type number*

poolname
value **allow-hint**

Configuring a Binding Database Agent for the Server Function

SUMMARY STEPS

- 1.
- 2.

[illegible]**rapid-commit**[illegible]

Configuring DHCP for IPv6 Address Assignment

-
-
- [, page 15](#) (required)

Prerequisites for Configuring DHCPv6 Address Assignment

- *vlan_id*

Enabling the DHCPv6 Server Function on an Interface



Note

Summary Steps

- 1.
 - 1.
 - 2.
 - 3.
 4. **link-address**
 5. **vendor-specific**
 6. **suboption** **address** **ascii** *ASCII-string* *hex-string*
- type number*
- poolname*

Detailed Steps

	Command	Purpose
Step 1		<ul style="list-style-type: none">
	Example:	
Step 1		
	Example:	
Step 2		
	Example:	
Step 3	<i>IPv6-prefix</i> <i>valid-lifetime preferred-lifetime</i>	
	link-address	
	Router(config-dhcpv6)# link-address 2001:1001::0/64	
	Router(config-dhcpv6)# vendor-specific 9	
	<i>ASCII-string</i> { <i>hex-string</i>	

<i>type number</i>	
<i>poolname</i> <i>value</i>	
Router# show ipv6 dhcp pool or Router# show ipv6 dhcp interface	
Router#copy running-config startup-config	



no ipv6 address dhcp client request

enable
configure terminal
interface
ipv6 address dhcp rapid-commit

Router> enable	
Router# configure terminal	
Router(config)# interface fastethernet 0/0	
[]	
Router(config-if)# ipv6 address dhcp rapid-commit	
ipv6 address dhcp client request vendor	
Router(config-if)# ipv6 dhcp client request vendor-specific	
Router(config-if)# end	
Router# show ipv6 dhcp interface	

Router> enable	
Router# configure terminal	
Router(config)# ipv6 dhcp pool dhcp-pool	
Router(config-dhcp) dns-server 2001:0DB8:3000:3000::42	
Router(config-dhcp)# domain-name domain1.com	
Router(config-dhcp)# exit	

Enabling Processing of Packets with Source Routing Header Options

SUMMARY STEPS

- 1.
- 2.
- 3.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">•
Step 2	Example:	
Step 3	Example:	

Configuring the DHCPv6 Server Options

-
-
-
-
-
-
-
-

Configuring the Information Refresh Server Option

SUMMARY STEPS

1.
2.
3.
4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<div>Example:</div>	<div>•</div>
Step 2	<div>Example:</div>	
Step 3	<div>Example:</div>	
Step 4	<div>Example:</div>	

Importing the Information Refresh Server Option

SUMMARY STEPS

1.
2.
3.
4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	

Configuring NIS- and NISP-Related Server Options

SUMMARY STEPS

-
-
-
-
-
-
-

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	
Step 5	Example:	
Step 6	Example:	
Step 7	Example:	

Importing NIS- and NIS+-Related Server Options

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.
- 5.

6.



	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	
Step 5	Example:	

Configuring the SNTP Server Option

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.
- 5.



DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	
Step 5	Example:	

Importing the SNTP Server Option

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	

Importing Stateless DHCPv6 Server Options

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.
- 5.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	

	Command or Action	Purpose
Step 3	Example:	
Step 4	Example:	
Step 5	Example:	

Defining a General Prefix with the DHCPv6 Prefix Delegation Client Function

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	•
Step 2	Example:	

	Command or Action	Purpose
Step 3	Example:	
Step 4	Example:	<ul style="list-style-type: none">

Restarting the DHCPv6 Client on an Interface

SUMMARY STEPS

- 1.
- 2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	

Deleting Automatic Client Bindings from the DHCPv6 Binding Table

SUMMARY STEPS

- 1.
- 2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	

Troubleshooting DHCPv6

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	
Step 3	Example:	
Step 4	Example:	

Verifying DHCPv6 Configuration and Operation

SUMMARY STEPS

1.
2.
3.
4.
5.
6.
7.

DETAILED STEPS

	Command or Action	Purpose
Step 1	<div>Example:</div>	<div>•</div>
Step 2	<div>Example:</div>	
Step 3	<div>Example:</div>	
Step 4		
Step 5	<div>Example:</div>	

[]	
Router# show ipv6 dhcp pool	
Router# show running-config	

Examples

-
-
-
-
-

Sample Output from the show ipv6 dhcp Command

This device's DHCPv6 unique identifier(DUID): 000300010002FCA5DC1C

```
Router#
Client: FE80::202:FCFF:FEA5:DC39 (Ethernet2/1)
DUID: 000300010002FCA5DC1C
IA PD: IA ID 0x00040001, T1 0, T2 0
Prefix: 3FFE:C00:C18:11::/68
        preferred lifetime 180, valid lifetime 12345
        expires at Nov 08 2002 02:24 PM (12320 seconds)
Client: FE80::202:FCFF:FEA5:C039 (Ethernet2/1)
DUID: 000300010002FCA5C01C
IA PD: IA ID 0x00040001, T1 0, T2 0
Prefix: 3FFE:C00:C18:1::/72
        preferred lifetime 240, valid lifetime 54321
        expires at Nov 09 2002 02:02 AM (54246 seconds)
Prefix: 3FFE:C00:C18:2::/72
        preferred lifetime 300, valid lifetime 54333
        expires at Nov 09 2002 02:03 AM (54258 seconds)
Prefix: 3FFE:C00:C18:3::/72
        preferred lifetime 280, valid lifetime 51111
```

Router#

```
Database agent tftp://172.19.216.133/db.tftp:
  write delay: 69 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
    write timer expires in 56 seconds
  last read at Jan 06 2003 05:41 PM
  successful read times 1
  failed read times 0
  successful write times 3172
  failed write times 2
Database agent nvram:/dhcpv6-binding:
  write delay: 60 seconds, transfer timeout: 300 seconds
  last written at Jan 09 2003 01:54 PM,
    write timer expires in 37 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 3325
  failed write times 0
Database agent flash:/dhcpv6-db:
  write delay: 82 seconds, transfer timeout: 3 seconds
  last written at Jan 09 2003 01:54 PM,
    write timer expires in 50 seconds
  last read at never
  successful read times 0
  failed read times 0
  successful write times 2220
  failed write times 614
```

Router1#

```
Ethernet2/1 is in server mode
  Using pool: svr-pl
  Preference value: 20
  Rapid-Commit is disabled
```

Router2#

```
Ethernet2/1 is in client mode
  State is OPEN (1)
  List of known servers:
    Address: FE80::202:FCFF:FEA1:7439, DUID 000300010002FCA17400
    Preference: 20
      IA PD: IA ID 0x00040001, T1 120, T2 192
        Prefix: 3FFE:C00:C18:1::/72
          preferred lifetime 240, valid lifetime 54321
          expires at Nov 08 2002 09:10 AM (54319 seconds)
        Prefix: 3FFE:C00:C18:2::/72
          preferred lifetime 300, valid lifetime 54333
          expires at Nov 08 2002 09:11 AM (54331 seconds)
        Prefix: 3FFE:C00:C18:3::/72
          preferred lifetime 280, valid lifetime 51111
          expires at Nov 08 2002 08:17 AM (51109 seconds)
      DNS server: 2001:0DB8:1001::1
      DNS server: 2001:0DB8:1001::2
      Domain name: example1.net
      Domain name: example2.net
```

Sample Output from the show ipv6 dhcp pool Command

```

IA PD: IA ID not specified; being used by 00040001
  Prefix: 3FFE:C00:C18:1::/72
    preferred lifetime 240, valid lifetime 54321
  Prefix: 3FFE:C00:C18:2::/72
    preferred lifetime 300, valid lifetime 54333
  Prefix: 3FFE:C00:C18:3::/72
    preferred lifetime 280, valid lifetime 51111
Prefix from pool: local-p1, Valid lifetime 12345, Preferred lifetime 180
DNS server: 2001:0DB8:1001::1
DNS server: 2001:0DB8:1001::2
Domain name: example1.net
Domain name: example2.net
Domain name: example3.net
Active clients: 2
Current configuration : 22324 bytes
!
! Last configuration change at 14:59:38 PST Tue Jan 16 2001
! NVRAM config last updated at 04:25:39 PST Tue Jan 16 2001 by bird
!
hostname Router
!
ip cef
ipv6 unicast-routing
ipv6 cef
ipv6 cef accounting prefix-length
!
!
interface Ethernet0
 ip address 10.4.9.11 255.0.0.0
 media-type 10BaseT
 ipv6 address 2001:0DB8:C18:1::/64 eui-64

```

Configuration Examples for Implementing DHCP for IPv6

-
-
-
-

Configuring the DHCPv6 Server Function: Example

Configuring the DHCPv6 Client Function: Example

Configuring a Database Agent for the Server Function: Example

Configuring DHCP for IPv6 Address Assignment


```
suboption 2 ascii "IP-Phone"  
end
```

Configuring the Stateless DHCPv6 Function: Example

```
ipv6 dhcp pool dhcp-pool  
  dns-server 2001:0DB8:A:B::1  
  dns-server 2001:0DB8:3000:3000::42  
  domain-name example.com  
!  
interface Ethernet0/0  
  description Access link down to customers  
  ipv6 address 2001:0DB8:1234:42::1/64  
  ipv6 nd other-config-flag  
  ipv6 dhcp server dhcp-pool
```

-

-

Additional References

Related Documents

Related Topic	Document Title
	<i>Cisco IOS IPv6 Configuration Guide</i>
	<i>IPv6 Configuration Guide</i> <i>Cisco IOS</i>
	<i>Cisco IOS IPv6 Configuration Guide</i>
	<i>Cisco IOS IPv6 Configuration Guide</i>
	<i>Cisco IOS IPv6 Command Reference</i>

Standards

Standards	Title

MIBs

MIBs	MIBs Link

RFCs	Title
	<i>Dynamic Host Configuration Protocol for IPv6</i>
	<i>Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers</i>
	<i>IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) Version 6</i>
	<i>DNS Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i>
	<i>Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i>
	<i>Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6</i>
	<i>Information Refresh Time Option for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)</i>
	<i>Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option</i>

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*

dns-server (IPv6)
domain-name (IPv6)
import dns-server
import domain-name
import information refresh
import nis address
import nis domain-name
import nisp address
import nisp domain-name
import sip address
import sip domain-name
import sntp address
information refresh
ipv6 address autoconfig
ipv6 address dhcp
ipv6 address dhcp client request
ipv6 dhcp client information refresh minimum
ipv6 dhcp client pd
ipv6 dhcp database
ipv6 dhcp ping packets
ipv6 dhcp pool
ipv6 dhcp relay destination
ipv6 dhcp server
ipv6 nd other-config-flag
ipv6 source-route
nis address
nis domain-name
nisp address
nisp domain-name
prefix-delegation
prefix-delegation pool
show ipv6 dhcp
show ipv6 dhcp binding

show ipv6 dhcp conflict
show ipv6 dhcp database
show ipv6 dhcp interface
show ipv6 dhcp pool
show ipv6 interface
sntp address

Feature Information for Implementing DHCP for IPv6

[Start Here: Cisco IOS](#)

[Software Release Specifies for IPv6 Features](#)



Feature Information for Implementing DHCP for IPv6

Feature Name	Releases	Feature Information
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">

Feature Information for Implementing DHCP for IPv6 (continued)

		<ul style="list-style-type: none">
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">

DHCPv6 Ethernet Remote ID option	12.2(33)SRC 12.2(33)SXI Cisco IOS XE Release 2.1	This feature adds the remote-ID option to relayed (RELAY-FORWARD) DHCPv6 packets. The following section provides information about this feature: DHCPv6 Relay Options: Remote-ID for Ethernet Interfaces, page 8
DHCPv6—Relay—Reload Persistent Interface ID Option	12.2(33)SB 12.2(33)SRC 12.2(33)SXI Cisco IOS XE Release 2.1	This feature makes the interface-ID option, which is used by relay agents to decide which interface should be used when forwarding a RELAY-REPLY packet, persistent. The following section provides information about this feature: DHCPv6 Relay Options: Reload Persistent Interface-ID, page 8
DHCP—DHCPv6 Individual Address Assignment	12.4(24)T	This feature manages nonduplicate address assignment in the correct prefix based on the network where the host is connected. The following section provides information about this feature: DHCP for IPv6 Address Assignment, page 4 Configuring DHCP for IPv6 Address Assignment, page 13

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006–2009 Cisco Systems, Inc. All rights reserved.



Implementing Dynamic Multipoint VPN for IPv6

First Published: July 11, 2008

Last Updated: July 11, 2008

This document describes how to implement Dynamic Multipoint VPN for IPv6 feature, which allows users to better scale large and small IPsec Virtual Private Networks (VPNs) by combining generic routing encapsulation (GRE) tunnels, IP security (IPsec) encryption, and the Next Hop Resolution Protocol (NHRP). In Dynamic Multipoint Virtual Private Network (DMVPN) for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing DMVPN for IPv6”](#) section on page 26.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing DMVPN for IPv6, page 2](#)
- [Restrictions for Implementing DMVPN for IPv6, page 2](#)
- [Information About Implementing DMVPN for IPv6, page 2](#)
- [How to Configure DMVPN for IPv6, page 4](#)
- [Configuration Examples for Implementing DMVPN for IPv6, page 18](#)
- [Additional References, page 23](#)
- [Command Reference, page 24](#)
- [Feature Information for Implementing DMVPN for IPv6, page 26](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing DMVPN for IPv6

- This document assumes that you are familiar with IPv6 and IPv4. See the publications referenced in the [“Additional References”](#) section for IPv6 and IPv4 configuration and command reference information.
- Perform basic IPv6 addressing and basic connectivity as described in [“Implementing IPv6 Addressing and Basic Connectivity.”](#)
- Supported routing protocols include Border Gateway Protocol (BGP), Enhanced Interior Gateway Routing Protocol (EIGRP), On-Demand Routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP). One of these protocols must be enabled for DMVPN for IPv6 to work.

Restrictions for Implementing DMVPN for IPv6

- IPv6 can be configured only on the protected network.
- Every IPv6 NHRP interface is configured with one IPv6 unicast address. This address can be a globally reachable or unique local address.
- Every IPv6 NHRP interface is configured with one IPv6 link-local address that is unique across all the DMVPN hosts in the DMVPN cloud (that is, the hubs and spokes).
- IPv6 VRFs are not supported fully by IPv6 routing protocols such as EIGRP or OSPF. Therefore, DMVPN for IPv6 does not support IPv6 VRFs.
- The WAN network has to be a IPv4 network.

Information About Implementing DMVPN for IPv6

To configure DMVPN for Cisco IOS software, you must understand the following concept:

- [DMVPN for IPv6 Overview, page 2](#)

DMVPN for IPv6 Overview

The DMVPN feature combines NHRP routing, multipoint generic routing encapsulation (mGRE) tunnels, and IPsec encryption to provide users an ease of configuration via crypto profiles—which override the requirement for defining static crypto maps—and dynamic discovery of tunnel endpoints.

This feature relies on the following Cisco enhanced standard technologies:

- NHRP—A client and server protocol where the hub is the server and the spokes are the clients. The hub maintains an NHRP database of the public interface addresses of each spoke. Each spoke registers its real address when it boots and queries the NHRP database for real addresses of the destination spokes to build direct tunnels.
- mGRE tunnel interface—An mGRE tunnel interface allows a single GRE interface to support multiple IPsec tunnels and simplifies the size and complexity of the configuration.
- IPsec encryption—An IPsec tunnel interface allows for the protection of site-to-site IPv6 traffic with native encapsulation.

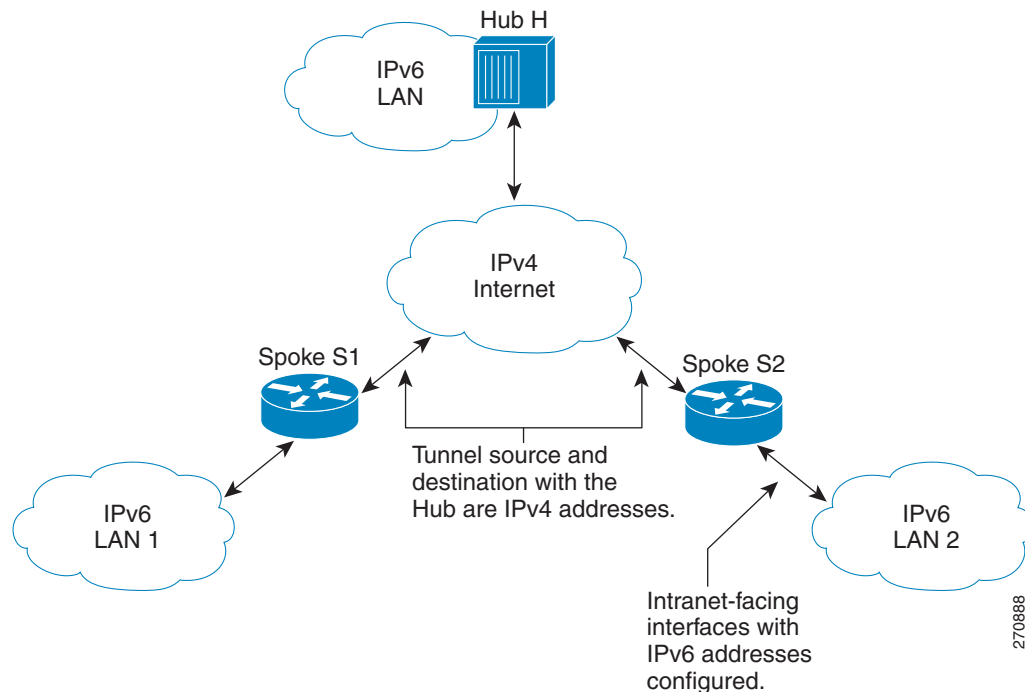
In DMVPN for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable. The intranets could be a mix of IPv4 or IPv6 clouds connected to each other using the DMVPN technologies, with the underlying carrier being traditional IPv4.

NHRP Routing

The NHRP protocol resolves a given intranet address (IPv4 or IPv6) to an Internet address (IPv4 nonbroadcast multiaccess [NBMA] address).

In [Figure 1](#), the intranets that are connected over the DMVPN network are IPv6 clouds, and the Internet is a pure IPv4 cloud. Spokes S1 and S2 are connected to the Hub H over the Internet using a statically configured mGRE tunnel. The address of the tunnel itself is in the IPv6 domain, because it is another node on the intranet. The source and destinations of the tunnel (the mGRE endpoints), however, are always in IPv4, in the Internet domain. The mGRE tunnel is aware of the IPv6 network because the GRE passenger protocol is an IPv6 packet, and the GRE transport (or carrier) protocol is an IPv4 packet.

Figure 1 IPv6 Topology That Triggers NHRP



When an IPv6 host in LAN L1 sends a packet destined to an IPv6 host in LAN L2, the packet is first routed to the gateway (which is Spoke S1) in LAN L1. Spoke S1 is a dual-stack router, which means both IPv4 and IPv6 are configured. The IPv6 routing table in S1 points to a next hop, which is the IPv6 address of the tunnel on Spoke S2. This is a VPN address that must be mapped to an NBMA address, triggering NHRP.

IPv6 NHRP Redirect and Shortcut Features

When IPv6 NHRP redirect is enabled, NHRP examines every data packet in the output feature path. If the data packet enters and leaves on the same logical network, it sends an NHRP traffic indication message to the originator of the data packet. In NHRP, a logical network is identified by the NHRP network ID, which groups multiple physical interfaces into a single logical network.

When IPv6 NHRP shortcut is enabled, NHRP intercepts every data packet in the output feature path. It checks to see if there is an NHRP cache entry to the destination of the data packet and, if yes, it replaces the current output adjacency with the one present in the NHRP cache. The data packet is therefore switched out using the new adjacency provided by NHRP.

IPv6 Routing

NHRP is automatically invoked for mGRE tunnels carrying the IPv6 passenger protocol. When a packet is routed and the packet is sent to the switching path, NHRP looks up the given next hop and, if required, initiates an NHRP resolution query. If the resolution is successful, NHRP populates the tunnel endpoint database, which then populates the Cisco Express Forwarding adjacency table. The subsequent packets are Cisco Express Forwarding switched if Cisco Express Forwarding is enabled.

IPv6 Addressing and Restrictions

IPv6 allows multiple unicast addresses on a given IPv6 interface. IPv6 also allows special address types, such as anycast, multicast, link-local addresses, and unicast addresses.

DMVPN for IPv6 has the following addressing restrictions:

- Every IPv6 NHRP interface is configured with one IPv6 unicast address. This address can be a globally reachable or unique local address.
- Every IPv6 NHRP interface is configured with one IPv6 link-local address that is unique across all the DMVPN hosts in the DMVPN cloud (that is, the hubs and spokes).
 - If no other tunnels on the router are using the same tunnel source, then the tunnel source address can be embedded into an IPv6 address.
 - If the router has only one DMVPN IPv6 tunnel, then manual configuration of the IPv6 link-local address is not required. Instead, use the **ipv6 enable** command to autogenerate a link-local address.
 - If the router has more than one DMVPN IPv6 tunnel, then the link-local address must be manually configured using the **ipv6 address fe80::2001 link-local** command.

How to Configure DMVPN for IPv6

To enable mGRE and IPsec tunneling for hub and spoke routers, you must configure an IPsec profile that uses a global IPsec policy template and configure your mGRE tunnel for IPsec encryption. This section contains the following procedures:

- [Configuring an IPsec Profile in DMVPN for IPv6, page 5](#) (required)
- [Configuring the Hub for IPv6 over DMVPN, page 6](#) (required)
- [Configuring the Spoke for IPv6 over DMVPN, page 9](#) (required)
- [Verifying DMVPN for IPv6 Configuration, page 12](#) (optional)
- [Monitoring and Maintaining DMVPN for IPv6 Configuration and Operation, page 13](#) (optional)

Configuring an IPsec Profile in DMVPN for IPv6

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

Prerequisites

Before configuring an IPsec profile, you must do the following:

- Define a transform set by using the **crypto ipsec transform-set** command.
- Make sure that Internet Security Association Key Management Protocol (ISAKMP) is configured with default ISAKMP settings. For further information about default ISAKMP settings, see the [Implementing IPsec in IPv6 Security](#) module and the [Cisco IOS IPv6 Command Reference](#).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto identity** *name*
4. **crypto ipsec profile** *name*
5. **set transform-set** *transform-set-name*
6. **set identity**
7. **set security-association lifetime** {*seconds seconds* | *kilobytes kilobytes*}
8. **set pfs** [*group1* | *group2*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	crypto identity <i>name</i>	Configures the identity of the router with a given list of distinguished names (DNs) in the certificate of the router.
	Example: Router(config)# crypto identity router1	

	Command or Action	Purpose
Step 4	crypto ipsec profile <i>name</i> Example: Router(config)# crypto ipsec profile example1	Defines the IPsec parameters that are to be used for IPsec encryption between “spoke and hub” and “spoke and spoke” routers. This command places the router in crypto map configuration mode.
Step 5	set transform-set <i>transform-set-name</i> Example: Router(config-crypto-map)# set transform-set example-set	Specifies which transform sets can be used with the IPsec profile.
Step 6	set identity Example: Router(config-crypto-map)# set identity router1	(Optional) Specifies identity restrictions to be used with the IPsec profile.
Step 7	set security-association lifetime { seconds <i>seconds</i> kilobytes <i>kilobytes</i> } Example: Router(config-crypto-map)# set security-association lifetime seconds 1800	(Optional) Overrides the global lifetime value for the IPsec profile.
Step 8	set pfs [group1 group2] Example: Router(config-crypto-map)# set pfs group2	(Optional) Specifies that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this IPsec profile.

Configuring the Hub for IPv6 over DMVPN

This task describes how to configure the hub router for IPv6 over DMVPN for mGRE and IPsec integration (that is, associate the tunnel with the IPsec profile configured in the previous procedure).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
5. **ipv6 address** *ipv6-address/prefix-length* **link-local**
6. **ipv6 mtu** *bytes*
7. **ipv6 nhrp authentication** *string*
8. **ipv6 nhrp map multicast dynamic**
9. **ipv6 nhrp network-id** *network-id*
10. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}

11. **tunnel mode** {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ipv6 | ipip [decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbsep}
12. **tunnel protection ipsec profile** *name* [shared]
13. **bandwidth** {interzone | total | session} {default | zone *zone-name*} *bandwidth-size*
14. **ipv6 nhrp holdtime** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 5	Configures a tunnel interface and enters interface configuration mode. The number argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if)# ipv6 address 2001:0db8:1:1::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	ipv6 address <i>ipv6-address/prefix-length</i> link-local Example: Router(config-if)# ipv6 address fe80::2001 link-local	Configures an IPv6 link-local address for an interface and enable IPv6 processing on the interface. A unique IPv6 link local address (across all DMVPN nodes in a DMVPN network) must be configured.
Step 6	ipv6 mtu <i>bytes</i> Example: Router(config-if)# ipv6 mtu 1400	Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface.
Step 7	ipv6 nhrp authentication <i>string</i> Example: Router(config-if)# ipv6 nhrp authentication examplexx	Configures the authentication string for an interface using the NHRP. Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.

	Command or Action	Purpose
Step 8	ipv6 nhrp map multicast dynamic Example: Router(config-if)# ipv6 nhrp map multicast dynamic	Allows NHRP to automatically add routers to the multicast NHRP mappings.
Step 9	ipv6 nhrp network-id network-id Example: Router(config-if)# ipv6 nhrp network-id 99	Enables the NHRP on an interface.
Step 10	tunnel source {ip-address ipv6-address interface-type interface-number} Example: Router(config-if)# tunnel source ethernet 0	Sets the source address for a tunnel interface.
Step 11	tunnel mode {aurp cayman dvmrp eon gre gre multipoint gre ipv6 ipip [decapsulate-any] ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbscp} Example: Router(config-if)# tunnel mode gre multipoint	Sets the encapsulation mode to mGRE for the tunnel interface.
Step 12	tunnel protection ipsec profile name [shared] Example: Router(config-if)# tunnel protection ipsec profile example_profile	Associates a tunnel interface with an IPsec profile. <ul style="list-style-type: none"> The name argument specifies the name of the IPsec profile; this value must match the name specified in the crypto ipsec profile name command.
Step 13	bandwidth {interzone total session} {default zone zone-name} bandwidth-size Example: Router(config-if)# bandwidth total 1200	Sets the current bandwidth value for an interface to higher-level protocols. <ul style="list-style-type: none"> The <i>kb/s</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommended bandwidth value is 1000 or greater.
Step 14	ipv6 nhrp holdtime seconds Example: Router(config-if)# ipv6 nhrp holdtime 3600	Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses.

Configuring the NHRP Redirect and Shortcut Features On the Hub

This task describes how to configure the NHRP redirect and shortcut features on the hub.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel number**
4. **ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}**

5. **ipv6 nhrp redirect** [timeout *seconds*]
6. **ipv6 nhrp shortcut**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 5	Configures a tunnel interface and enters interface configuration mode. The number argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if)# ipv6 address 2001:0db8:1:1::72/64	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.
Step 5	ipv6 nhrp redirect [timeout <i>seconds</i>] Example: Router(config-if)# ipv6 nhrp redirect	Enables NHRP redirect.
Step 6	ipv6 nhrp shortcut Example: Router(config-if)# ipv6 nhrp shortcut	Enables NHRP shortcut switching.

Configuring the Spoke for IPv6 over DMVPN

This task describes how to configure the spoke for IPv6 over DMVPN.


SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
5. **ipv6 address** *ipv6-address/prefix-length* **link-local**

6. **ipv6 mtu** *bytes*
7. **ipv6 nhrp authentication** *string*
8. **ipv6 nhrp map** *ipv6-address nbma-address*
9. **ipv6 nhrp map multicast** *ipv4-nbma-address*
10. **ipv6 nhrp nhs** *ipv6-nhs-address [net-address]*
11. **ipv6 nhrp network-id** *network-id*
12. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
13. **tunnel mode** {*aurp* | *cayman* | *dvmrp* | *eon* | **gre** | **gre multipoint** | **gre ipv6** | **ipip** | *decapsulate-any*} | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbscp**}
- or
- tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
14. **tunnel protection ipsec profile** *name* [**shared**]
15. **bandwidth** {**interzone** | **total** | **session**} {**default** | **zone** *zone-name*} *bandwidth-size*
16. **ipv6 nhrp holdtime** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>number</i> Example: Router(config)# interface tunnel 5	Configures a tunnel interface and enters interface configuration mode The <i>number</i> argument specifies the number of the tunnel interfaces that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if) ipv6 address 2001:0db8:1:1::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	ipv6 address <i>ipv6-address/prefix-length</i> link-local Example: Router(config-if)# ipv6 address fe80::2001 link-local	Configures an IPv6 link-local address for an interface and enables IPv6 processing on the interface. A unique IPv6 link local address (across all DMVPN nodes in a DMVPN network) must be configured.

	Command or Action	Purpose
Step 6	ipv6 mtu bytes Example: Router(config-if)# ipv6 mtu 1400	Sets the MTU size of IPv6 packets sent on an interface.
Step 7	ipv6 nhrp authentication string Example: Router(config-if)# ipv6 nhrp authentication examplexx	Configures the authentication string for an interface using the NHRP. Note The NHRP authentication string must be set to the same value on all hubs and spokes that are in the same DMVPN network.
Step 8	ipv6 nhrp map ipv6-address nbma-address Example: Router(config-if)# ipv6 nhrp map 2001:0DB8:3333:4::5 10.1.1.1	Statically configures the IPv6-to-NBMA address mapping of IPv6 destinations connected to an NBMA network.  Note Only IPv4 NBMA addresses are supported, not ATM or Ethernet addresses.
Step 9	ipv6 nhrp map multicast ipv4-nbma-address Example: Router(config-if)# ipv6 nhrp map multicast 10.11.11.99	Maps destination IPv6 addresses to IPv4 NBMA addresses.
Step 10	ipv6 nhrp nhs ipv6-nhs-address [net-address] Example: Router(config-if)# ipv6 nhrp nhs 2001:0DB8:3333:4::5 2001:0DB8::/64	Specifies the address of one or more IPv6 NHRP servers.
Step 11	ipv6 nhrp network-id network-id Example: Router(config-if)# ipv6 nhrp network-id 99	Enables the NHRP on an interface.
Step 12	tunnel source {ip-address ipv6-address interface-type interface-number} Example: Router(config-if)# tunnel source ethernet 0	Sets the source address for a tunnel interface.

	Command or Action	Purpose
Step 13	<pre>tunnel mode {aurp cayman dvmrp eon gre gre multipoint gre ipv6 ipip [decapsulate-any] ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbscp} or tunnel destination {host-name ip-address ipv6-address}</pre> <p>Example: Router(config-if)# tunnel mode gre multipoint</p> <p>or Router(config-if)# tunnel destination 10.1.1.1</p>	<p>Sets the encapsulation mode to mGRE for the tunnel interface.</p> <p>Use this command if data traffic can use dynamic spoke-to-spoke traffic.</p> <p>or</p> <p>Specifies the destination for a tunnel interface.</p> <p>Use this command if data traffic can use hub-and-spoke tunnels.</p>
Step 14	<pre>tunnel protection ipsec profile name [shared]</pre> <p>Example: Router(config-if)# tunnel protection ipsec profile example1</p>	<p>Associates a tunnel interface with an IPsec profile.</p> <ul style="list-style-type: none"> The <i>name</i> argument specifies the name of the IPsec profile; this value must match the name specified in the crypto ipsec profile name command.
Step 15	<pre>bandwidth {interzone total session} {default zone zone-name} bandwidth-size</pre> <p>Example: Router(config-if)# bandwidth total 1200</p>	<p>Sets the current bandwidth value for an interface to higher-level protocols.</p> <ul style="list-style-type: none"> The <i>kb/s</i> argument specifies the bandwidth in kilobits per second. The default value is 9. The recommended bandwidth value is 1000 or greater. The bandwidth setting for the spoke need not equal the bandwidth setting for the DMVPN hub. It is usually easier if all of the spokes use the same or similar value.
Step 16	<pre>ipv6 nhrp holdtime seconds</pre> <p>Example: Router(config-if)# ipv6 nhrp holdtime 3600</p>	<p>Changes the number of seconds that NHRP NBMA addresses are advertised as valid in authoritative NHRP responses.</p>

Verifying DMVPN for IPv6 Configuration

This optional task describes how to display information to verify DMVPN for IPv6 configuration. Use the following optional commands as needed to verify configuration and operation.

SUMMARY STEPS

1. **enable**
2. **show dmvpn** [ipv4 | ipv6] [peer [nbma | tunnel {ip-address | ipv6-address}]] [network {ip-address mask}] [vrf vrf-name] [interface tunnel number] [detail] [static] [debug-condition]
3. **show ipv6 nhrp** [dynamic [ipv6-address] | incomplete | static] [address | interface] [brief | detail] [purge]
4. **show ipv6 nhrp multicast** [ipv6-address | interface]
5. **show ipv6 nhrp summary**

6. show ipv6 nhrp traffic [interface tunnel number]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show dmvpn [ipv4 ipv6] [peer [nbma tunnel {ip-address ipv6-address}] network {ip-address mask}] [vrf vrf-name] [interface tunnel number] [detail] [static] [debug-condition] Example: Router# show dmvpn 2001:0db8:1:1::72/64	Displays DMVPN-specific session information.
Step 3	show ipv6 nhrp [dynamic [ipv6-address] incomplete static] [address interface] [brief detail] [purge] Example: Router# show ipv6 nhrp	Displays NHRP mapping information.
Step 4	show ipv6 nhrp multicast [ipv6-address interface] Example: Router# show ipv6 nhrp multicast	Displays NHRP multicast mapping information.
Step 5	show ipv6 nhrp summary Example: Router# show ipv6 nhrp summary	Displays NHRP mapping summary information.
Step 6	show ipv6 nhrp traffic [interface tunnel number] Example: Router# show ipv6 nhrp traffic	Displays NHRP traffic statistics information.

Monitoring and Maintaining DMVPN for IPv6 Configuration and Operation

This optional task explains how to display information to monitor and maintain DMVPN for IPv6 configuration and operation. Use the following optional commands only as needed to monitor configuration and operation.

SUMMARY STEPS

1. enable

2. **clear dmvpn session** [peer {nbma | tunnel *ipv4-address* | *ipv6-address*}] [interface tunnel *number*] [vrf *vrf-name*] [static]
3. **clear ipv6 nhrp** [*ipv6-address* | counters]
4. **debug dmvpn** [condition [unmatched] | [peer [nbma | tunnel | *ipv4-address* | *ipv6-address*] | vrf [*vrf-name*] | interface {tunnel *number*} | error | detail | packet | all | nhrp [crypto | tunnel | socket | all]]
5. **debug nhrp** {ipv4 | ipv6} [cache | extension | packet | rate]
6. **debug nhrp condition** [peer [nbma | tunnel | *ip-address* | *ipv6-address*]] | interface tunnel *number* | [vrf *vrf-name*]
7. **debug nhrp** {ipv4 | ipv6} error

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear dmvpn session [peer {nbma tunnel <i>ipv4-address</i> <i>ipv6-address</i> }] [interface tunnel <i>number</i>] [vrf <i>vrf-name</i>] [static] Example: Router# clear dmvpn session	Clears DMVPN sessions.
Step 3	clear ipv6 nhrp [<i>ipv6-address</i> counters] Example: Router# clear ipv6 nhrp	Clears all dynamic entries from the NHRP cache.
Step 4	debug dmvpn [condition [unmatched] [peer [nbma tunnel <i>ipv4-address</i> <i>ipv6-address</i>] vrf [<i>vrf-name</i>] interface {tunnel <i>number</i> } error detail packet all nhrp [crypto tunnel socket all]] Example: Router# debug dmvpn	Displays debug DMVPN session information.
Step 5	debug nhrp {ipv4 ipv6} [cache extension packet rate] Example: Router# debug nhrp ipv6	Enable NHRP debugging.

	Command or Action	Purpose
Step 6	<pre>debug nhrp condition [peer [nbma tunnel ip-address ipv6-address]] interface tunnel number [vrf vrf-name]</pre> <p>Example: Router# debug nhrp condition</p>	Enables NHRP conditional debugging.
Step 7	<pre>debug nhrp {ipv4 ipv6} error</pre> <p>Example: Router# debug nhrp ipv6 error</p>	Displays NHRP error level debugging information.

Examples

This section provides the following output examples:

- [Sample Output from the show dmvpn Command, page 15](#)
- [Sample Output from the show ipv6 nhrp Command, page 17](#)
- [Sample Output for the debug nhrp Command, page 18](#)

Sample Output from the show dmvpn Command

The following sample output is from the **show dmvpn** command, with the **ipv6** and **detail** keywords, for the hub:

```
Router# show dmvpn ipv6 detail
```

```
Legend: Attrib --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface Tunnel1 is up/up, Addr. is 10.0.0.3, VRF ""
  Tunnel Src./Dest. addr: 192.169.2.9/MGRE, Tunnel VRF ""
  Protocol/Transport: "multi-GRE/IP", Protect "test_profile"
Type:Hub, Total NBMA Peers (v4/v6): 2
  1.Peer NBMA Address: 192.169.2.10
    Tunnel IPv6 Address: 2001::4
    IPv6 Target Network: 2001::4/128
    # Ent: 2, Status: UP, UpDn Time: 00:01:51, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  2.Peer NBMA Address: 192.169.2.10
    Tunnel IPv6 Address: 2001::4
    IPv6 Target Network: FE80::2/128
    # Ent: 0, Status: UP, UpDn Time: 00:01:51, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  3.Peer NBMA Address: 192.169.2.11
    Tunnel IPv6 Address: 2001::5
    IPv6 Target Network: 2001::5/128
    # Ent: 2, Status: UP, UpDn Time: 00:26:38, Cache Attrib: D
Type:Hub, Total NBMA Peers (v4/v6): 2
  4.Peer NBMA Address: 192.169.2.11
    Tunnel IPv6 Address: 2001::5
    IPv6 Target Network: FE80::3/128
    # Ent: 0, Status: UP, UpDn Time: 00:26:38, Cache Attrib: D
Pending DMVPN Sessions:
```

```

Interface: Tunnell
IKE SA: local 192.169.2.9/500 remote 192.169.2.10/500 Active
Crypto Session Status: UP-ACTIVE
fvrf: (none), Phase1_id: 192.169.2.10
IPSEC FLOW: permit 47 host 192.169.2.9 host 192.169.2.10
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0x BB0ED02, transform : esp-3des esp-sha-hmac
    Socket State: Open

Interface: Tunnell
IKE SA: local 192.169.2.9/500 remote 192.169.2.11/500 Active
Crypto Session Status: UP-ACTIVE
fvrf: (none), Phase1_id: 192.169.2.11
IPSEC FLOW: permit 47 host 192.169.2.9 host 192.169.2.11
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0xB79B277B, transform : esp-3des esp-sha-hmac
    Socket State: Open

```

The following sample output is from the **show dmvpn** command, with the **ipv6** and **detail** keywords, for the spoke:

```

Router# show dmvpn ipv6 detail

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete
        N - NATed, L - Local, X - No Socket
        # Ent --> Number of NHRP entries with same NBMA peer
        NHS Status: E --> Expecting Replies, R --> Responding
        UpDn Time --> Up or Down Time for a Tunnel
=====

Interface Tunnell is up/up, Addr. is 10.0.0.1, VRF ""
    Tunnel Src./Dest. addr: 192.169.2.10/MGRE, Tunnel VRF ""
    Protocol/Transport: "multi-GRE/IP", Protect "test_profile"

IPv6 NHS: 2001::6 RE
Type:Spoke, Total NBMA Peers (v4/v6): 1
    1.Peer NBMA Address: 192.169.2.9
        Tunnel IPv6 Address: 2001::6
        IPv6 Target Network: 2001::/112
        # Ent: 2, Status: NHRP, UpDn Time: never, Cache Attrib: S

IPv6 NHS: 2001::6 RE
Type:Unknown, Total NBMA Peers (v4/v6): 1
    2.Peer NBMA Address: 192.169.2.9
        Tunnel IPv6 Address: FE80::1
        IPv6 Target Network: FE80::1/128
        # Ent: 0, Status: UP, UpDn Time: 00:00:24, Cache Attrib: D

Pending DMVPN Sessions:

Interface: Tunnell
IKE SA: local 192.169.2.10/500 remote 192.169.2.9/500 Active
Crypto Session Status: UP-ACTIVE
fvrf: (none), Phase1_id: 192.169.2.9
IPSEC FLOW: permit 47 host 192.169.2.10 host 192.169.2.9
    Active SAs: 2, origin: crypto map
    Outbound SPI : 0x6F75C431, transform : esp-3des esp-sha-hmac
    Socket State: Open

```


Sample Output from the show ipv6 nhrp Command

The following sample output is from the **show ipv6 nhrp** command for the hub and the spoke:

Hub

Router# **show ipv6 nhrp**

```
2001::4/128 via 2001::4
  Tunnel1 created 00:02:40, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.10
2001::5/128 via 2001::5
  Tunnel1 created 00:02:37, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.11
FE80::2/128 via 2001::4
  Tunnel1 created 00:02:40, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.10
FE80::3/128 via 2001::5
  Tunnel1 created 00:02:37, expire 00:00:47
  Type: dynamic, Flags: unique registered used
  NBMA address: 192.169.2.11
```

Spoke

Router# **show ipv6 nhrp**

```
2001::8/128
  Tunnel1 created 00:00:13, expire 00:02:51
  Type: incomplete, Flags: negative
  Cache hits: 2
2001::/112 via 2001::6
  Tunnel1 created 00:01:16, never expire
  Type: static, Flags: used
  NBMA address: 192.169.2.9
FE80::1/128 via FE80::1
  Tunnel1 created 00:01:15, expire 00:00:43
  Type: dynamic, Flags:
  NBMA address: 192.169.2.9
```

Sample Output from the show ipv6 nhrp multicast Command

The following sample output is from the **show ipv6 nhrp multicast** command for the hub and the spoke:

Hub

Router# **show ipv6 nhrp multicast**

I/F	NBMA address	
Tunnel1	192.169.2.10	Flags: dynamic
Tunnel1	192.169.2.11	Flags: dynamic

Spoke

Router# **show ipv6 nhrp multicast**

I/F	NBMA address	
Tunnel1	192.169.2.9	Flags: static

Sample Output for the show ipv6 nhrp traffic Command

The following sample output is from the **show ipv6 nhrp traffic** command:

Router# **show ipv6 nhrp traffic**

```
Tunnel0: Max-send limit:100Pkts/10Sec, Usage:0%
Sent: Total 8
1 Resolution Request 1 Resolution Reply 6 Registration Request
0 Registration Reply 0 Purge Request 0 Purge Reply
0 Error Indication 0 Traffic Indication
Rcvd: Total 5
1 Resolution Request 1 Resolution Reply 0 Registration Request
2 Registration Reply 0 Purge Request 0 Purge Reply
0 Error Indication 1 Traffic Indication
```

Sample Output for the debug nhrp Command

The following sample output is from the **debug nhrp** command with the **ipv6** keyword:

```
Router# debug nhrp ipv6

Aug  9 13:13:41.486: NHRP: Attempting to send packet via DEST
        - 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: Encapsulation succeeded.
Aug  9 13:13:41.486: NHRP: Tunnel NBMA addr 11.11.11.99
Aug  9 13:13:41.486: NHRP: Send Registration Request via Tunnel0 vrf 0, packet size: 105
Aug  9 13:13:41.486: src: 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32,
        dst: 2001:0db8:3c4d:0015:0000:0000:1a2f:3d2c/32
Aug  9 13:13:41.486: NHRP: 105 bytes out Tunnel0
Aug  9 13:13:41.486: NHRP: Receive Registration Reply via Tunnel0 vrf 0, packet size: 125
```

Configuration Examples for Implementing DMVPN for IPv6

This section provides the following DMVPN configuration examples:

- [Configuring an IPsec Profile: Example, page 18](#)
- [Configuring the Hub for DMVPN: Example, page 18](#)
- [Configuring the NHRP Redirect and Shortcut Features On the Hub: Example, page 20](#)
- [Configuring the Spoke for DMVPN: Example, page 21](#)

Configuring an IPsec Profile: Example

The following example shows how to configure an IPsec profile:

```
Router(config)# crypto identity router1
Router(config)# crypto ipsec profile example1
Router(config-crypto-map)# set transform-set example-set
Router(config-crypto-map)# set identity router1
Router(config-crypto-map)# set security-association lifetime seconds 1800
Router(config-crypto-map)# set pfs group2
```

Configuring the Hub for DMVPN: Example

This example displays information on configuring the hub for DMVPN:

```
Router# show running-config

version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
```

```
!
hostname Hub-99
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
clock timezone IST 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
.
.
.
ip cef
no ip domain lookup
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
!
archive
log config
hidekeys
!
!
crypto isakmp policy 1
authentication pre-share
crypto isakmp key cisco123 address 10.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set cisco-ts esp-3des esp-md5-hmac
mode transport
!
crypto ipsec profile cisco-ipsec
set transform-set cisco-ts
.
.
.
interface Tunnel0
bandwidth 100000
ip address 10.1.1.99 255.255.255.0
no ip redirects
ip nhrp map multicast dynamic
delay 50000
ipv6 address 2001:0DB8:99/64 2001:0db8::99/64
ipv6 address FE80::0B:0B:0B:8F link-local
ipv6 enable
ipv6 eigrp 1
no ipv6 split-horizon eigrp 1
no ipv6 next-hop-self eigrp 1
ipv6 nhrp map multicast dynamic
ipv6 nhrp network-id 99
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile cisco-ipsec
!
```

```

interface Ethernet0/0
ip address 10.11.11.99 255.255.255.0
!
interface Ethernet0/1
no ip address
shutdown
!
interface Ethernet0/2
no ip address
shutdown
interface Ethernet0/3
no ip address
shutdown
!
interface Ethernet1/0
no ip address
ipv6 address 2001:0db8:EEEE::99/64
ipv6 enable
ipv6 eigrp 1
!
interface Ethernet1/1
no ip address
shutdown
!
interface Ethernet1/2
no ip address
shutdown
!
interface Ethernet1/3
no ip address
shutdown
!
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
!
ipv6 router eigrp 1
no shutdown
!
control-plane
!
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
login
!
exception data-corruption buffer truncate

```

Configuring the NHRP Redirect and Shortcut Features On the Hub: Example

The following example shows how to configure NHRP redirect and shortcut features on the hub:

```

Router(config)# interface tunnel 5
Router(config-if)# ipv6 address 2001:0db8:1:1::72/64
Router(config-if)# ipv6 nhrp redirect
Router(config-if)# ipv6 nhrp shortcut

```

Configuring the Spoke for DMVPN: Example

This example provides **show running-config** output for a DMVPN spoke configuration:

```
Router# show running-config

version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Spoke-11
!
boot-start-marker
boot-end-marker
!
logging message-counter syslog
!
no aaa new-model
clock timezone IST 0
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
ip cef
ipv6 unicast-routing
ipv6 cef
!
multilink bundle-name authenticated
!
!
archive
  log config
  hidekeys
!
!
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key cisco123 address 10.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set cisco-ts esp-3des esp-md5-hmac
  mode transport
!
crypto ipsec profile cisco-ipsec
  set transform-set cisco-ts
!
interface Tunnel0
  bandwidth 100000
  no ip address
  no ip redirects
  delay 50000
  ipv6 address 2001:0DB8::11/64
  ipv6 address FE80::0B:0B:0B:0B link-local
  ipv6 eigrp 1
  no ipv6 split-horizon eigrp 1
  no ipv6 next-hop-self eigrp 1
  ipv6 nhrp map 2001:0db8::11/64 10.11.11.99
  ipv6 nhrp map multicast 10.11.11.99
```

```

ipv6 nhrp network-id 99
ipv6 nhrp nhs 2001:0db8::99
tunnel source Ethernet0/0
tunnel mode gre multipoint
tunnel protection ipsec profile cisco-ipsec
!
interface Ethernet0/0
ip address 10.11.11.11 255.255.255.0
ipv6 enable
ipv6 nd ra mtu suppress
!
interface Ethernet0/1
no ip address
!
interface Ethernet0/2
no ip address
shutdown
!
interface Ethernet0/3
no ip address
shutdown
!
interface Ethernet1/0
ip address 172.16.11.11 255.255.255.0
ipv6 address 2001:0db8:dddd::1/64
ipv6 enable
ipv6 nd ra mtu suppress
ipv6 eigrp 1
!
interface Ethernet1/1
no ip address
shutdown
ipv6 enable
ipv6 nd ra mtu suppress
!
interface Ethernet1/2
no ip address
shutdown
!
interface Ethernet1/3
no ip address
shutdown
!
ip forward-protocol nd
!
!
ip http server
no ip http secure-server
!
ipv6 router eigrp 1
no shutdown
!
control-plane
!
!
line con 0
exec-timeout 0 0
logging synchronous
line aux 0
line vty 0 4
login
!
exception data-corruption buffer truncate

```

Additional References

The following sections provide references related to the Implementing DMVPN for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 IPsec	“Implementing IPsec in IPv6 Security,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 basic connectivity	“Implementing IPv6 Addressing and Basic Connectivity,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
DMVPN implementation for IPv4	“Dynamic Multipoint VPN (DMVPN)” module of the <i>Cisco IOS Security Configuration Guide</i>
DMVPN commands for IPv4	Cisco IOS Security Command Reference
NHRP for IPv4	“Configuring NHRP” module of the <i>Cisco IOS IP Addressing Services Configuration Guide</i>
NHRP commands for IPv4	“NHRP Commands” section of the <i>Cisco IOS IP Addressing Services Command Reference</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2332	<i>NBMA Next Hop Resolution Protocol (NHRP)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **clear dmvpn session**
- **clear ipv6 nhrp**
- **crypto identity**
- **crypto ipsec profile**
- **debug dmvpn**
- **debug nhrp**
- **debug nhrp condition**
- **debug nhrp error**
- **ipv6 mtu**
- **ipv6 nhrp authentication**
- **ipv6 nhrp holdtime**
- **ipv6 nhrp interest**
- **ipv6 nhrp map**

- **ipv6 nhrp map multicast**
- **ipv6 nhrp map multicast dynamic**
- **ipv6 nhrp max-send**
- **ipv6 nhrp network-id**
- **ipv6 nhrp nhs**
- **ipv6 nhrp record**
- **ipv6 nhrp redirect**
- **ipv6 nhrp registration**
- **ipv6 nhrp responder**
- **ipv6 nhrp server-only**
- **ipv6 nhrp shortcut**
- **ipv6 nhrp trigger-svc**
- **ipv6 nhrp use**
- **set security-association lifetime**
- **set pfs**
- **set transform-set**
- **show dmvpn**
- **show ipv6 nhrp**
- **show ipv6 nhrp multicast**
- **show ipv6 nhrp summary**
- **show ipv6 nhrp traffic**

Feature Information for Implementing DMVPN for IPv6

Table 1 lists the release history for this feature.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing DMVPN for IPv6

Feature Name	Releases	Feature Information
DMVPN for IPv6	12.4(20)T	The Dynamic Multipoint VPN feature allows users to better scale large and small IPsec Virtual Private Networks by combining generic routing encapsulation tunnels, IPsec encryption, and NHRP. In DMVPN for IPv6, the public network (the Internet) is a pure IPv4 network, and the private network (the intranet) is IPv6 capable.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008–2009 Cisco Systems, Inc. All rights reserved.



Implementing EIGRP for IPv6

First Published: February 27, 2006

Last Updated: March 17, 2009

Customers can configure Enhanced Interior Gateway Routing Protocol (EIGRP) to route IPv6 prefixes. EIGRP IPv4 runs over an IPv4 transport, communicates only with IPv4 peers, and advertises only IPv4 routes, and EIGRP for IPv6 follows the same model. EIGRP for IPv4 and EIGRP for IPv6 are configured and managed separately. However, the configuration of EIGRP for IPv4 and IPv6 is similar and provides operational familiarity and continuity.

This document provides information about configuring and implementing EIGRP for IPv6.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing EIGRP for IPv6” section on page 25](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing EIGRP for IPv6, page 2](#)
- [Restrictions for Implementing EIGRP for IPv6, page 2](#)
- [Information About Implementing EIGRP for IPv6, page 2](#)
- [How to Implement EIGRP for IPv6, page 4](#)
- [Configuration Examples for Implementing EIGRP for IPv6, page 21](#)
- [Where to Go Next, page 22](#)
- [Additional References, page 22](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006–2009 Cisco Systems, Inc. All rights reserved.

- [Command Reference, page 23](#)
- [Feature Information for Implementing EIGRP for IPv6, page 25](#)

Prerequisites for Implementing EIGRP for IPv6

- This document assumes that you are familiar with EIGRP IPv4.
- This document assumes that users have a basic knowledge of IPv6 addressing.

Restrictions for Implementing EIGRP for IPv6

This section lists ways in which EIGRP for IPv6 differs from EIGRP IPv4 and lists EIGRP for IPv6 restrictions:

- EIGRP for IPv6 is directly configured on the interfaces over which it runs. This feature allows EIGRP for IPv6 to be configured without the use of a global IPv6 address. There is no network statement in EIGRP for IPv6.

In per-interface configuration at system startup, if EIGRP has been configured on an interface, then the EIGRP protocol may start running before any EIGRP router mode commands have been executed.

- An EIGRP for IPv6 protocol instance requires a router ID before it can start running.
- EIGRP for IPv6 has a shutdown feature. The routing process should be in “no shutdown” mode in order to start running.
- When a user uses a passive-interface configuration, EIGRP for IPv6 need not be configured on the interface that is made passive.
- EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list** command. Use of the **route-map** command is not supported for route filtering with a distribute list.

Information About Implementing EIGRP for IPv6

To configure EIGRP for IPv6 in Cisco IOS software, you should understand the following concepts:

- [Cisco EIGRP for IPv6 Implementation, page 2](#)

Cisco EIGRP for IPv6 Implementation

EIGRP is an enhanced version of the IGRP developed by Cisco. EIGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of EIGRP have improved substantially over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the diffusing update algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation and allows all devices involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

EIGRP provides the following features:

- Increased network width—With Routing Information Protocol (RIP), the largest possible width of your network is 15 hops. When EIGRP is enabled, the largest possible width is 224 hops. Because the EIGRP metric is large enough to support thousands of hops, the only barrier to expanding the network is the transport layer hop counter. Cisco works around this limitation by incrementing the transport control field only when an IPv4 or an IPv6 packet has traversed 15 routers and the next hop to the destination was learned by way of EIGRP. When a RIP route is being used as the next hop to the destination, the transport control field is incremented as usual.
- Fast convergence—The DUAL algorithm allows routing information to converge as quickly as any other routing protocol.
- Partial updates—EIGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for EIGRP packets.
- Neighbor discovery mechanism—This is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.
- Arbitrary route summarization.
- Scaling—EIGRP scales to large networks.
- Route filtering—EIGRP for IPv6 provides route filtering using the **distribute-list prefix-list** command. Use of the **route-map** command is not supported for route filtering with a distribute list.

EIGRP has the following four basic components:

- Neighbor discovery—Neighbor discovery is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. EIGRP neighbor discovery is achieved with low overhead by periodically sending small hello packets. EIGRP neighbors can also discover a neighbor that has recovered after an outage because the recovered neighbor will send out a hello packet. As long as hello packets are received, the Cisco IOS software can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring routers can exchange routing information.

**Note**

In Cisco IOS Release 12.2(33)SRD, the **no shutdown** command must be explicitly configured when enabling the EIGRP IPv6 router process. If the **no shutdown** command is not configured, IPv6 EIGRP neighbors will be lost.

- Reliable transport protocol—The reliable transport protocol is responsible for guaranteed, ordered delivery of EIGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some EIGRP packets must be sent reliably and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet) it is not necessary to send hello packets reliably to all neighbors individually. Therefore, EIGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, which is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when unacknowledged packets are pending. This provision helps to ensure that convergence time remains low in the presence of varying speed links.
- DUAL finite state machine—The DUAL finite state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses several metrics including distance and cost information to select efficient, loop-free paths. When multiple routes to a neighbor exist, DUAL determines which route has the lowest metric (named the feasible distance), and enters this route into the routing table. Other possible routes to this neighbor with larger metrics

are received, and DUAL determines the reported distance to this network. The reported distance is defined as the total metric advertised by an upstream neighbor for a path to a destination. DUAL compares the reported distance with the feasible distance, and if the reported distance is less than the feasible distance, DUAL considers the route to be a feasible successor and enters the route into the topology table. The feasible successor route that is reported with the lowest metric becomes the successor route to the current route if the current route fails. To avoid routing loops, DUAL ensures that the reported distance is always less than the feasible distance for a neighbor router to reach the destination network; otherwise, the route to the neighbor may loop back through the local router.

- Protocol-dependent modules—When there are no feasible successors to a route that has failed, but there are neighbors advertising the route, a recomputation must occur. This is the process where DUAL determines a new successor. The amount of time required to recompute the route affects the convergence time. Recomputation is processor-intensive; it is advantageous to avoid unneeded recomputation. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, DUAL will use them in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. An example is the EIGRP module, which is responsible for sending and receiving EIGRP packets that are encapsulated in IPv4 or IPv6. It is also responsible for parsing EIGRP packets and informing DUAL of the new information received. EIGRP asks DUAL to make routing decisions, but the results are stored in the IPv4 or IPv6 routing table. Also, EIGRP is responsible for redistributing routes learned by other IPv4 or IPv6 routing protocols.

How to Implement EIGRP for IPv6

The tasks required to implement EIGRP for IPv6 are described in the following sections:

- [Enabling EIGRP for IPv6 on an Interface, page 4](#)
- [Configuring the Percentage of Link Bandwidth Used by EIGRP, page 6](#)
- [Configuring Summary Aggregate Addresses, page 7](#)
- [Configuring EIGRP Route Authentication, page 8](#)
- [Instructing the Next Hop in EIGRP, page 10](#)
- [Adjusting the Interval Between Hello Packets in EIGRP for IPv6, page 11](#)
- [Adjusting the Hold Time in EIGRP for IPv6, page 12](#)
- [Disabling Split Horizon in EIGRP for IPv6, page 13](#)
- [Configuring EIGRP Stub Routing for Greater Network Stability, page 14](#)
- [Customizing an EIGRP for IPv6 Routing Process, page 16](#)
- [Monitoring and Maintaining EIGRP, page 19](#)

Enabling EIGRP for IPv6 on an Interface

Perform the following task to enable EIGRP for IPv6 on a specified interface. EIGRP for IPv6 is directly configured on the interfaces over which it runs, which allows EIGRP for IPv6 to be configured without the use of a global IPv6 address.

**Note**

In Cisco IOS Release 12.2(33)SRD, the **no shutdown** command must be explicitly configured as shown in step 10 of the this task. If the **no shutdown** command is not configured, IPv6 EIGRP neighbors will be lost.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 enable**
6. **ipv6 eigrp** *as-number*
7. **no shutdown**
8. **ipv6 router eigrp** *as-number*
9. **router-id** {*ip-address* | *ipv6-address*}
10. **no shutdown**
11. **exit**
12. **show ipv6 eigrp interfaces** [*type number*] [*as-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is to be configured.
Step 5	ipv6 enable Example: Router(config-if)# ipv6 enable	Enables IPv6 processing on an interface that has not been configured with an explicit IPv6 address.

	Command or Action	Purpose
Step 6	ipv6 eigrp <i>as-number</i> Example: Router(config-if)# ipv6 eigrp 1	Enables EIGRP for IPv6 on a specified interface.
Step 7	no shutdown Example: Router(config-if) no shutdown	Allows the user to start the EIGRP for IPv6 protocol without changing any per-interface configuration.
Step 8	ipv6 router eigrp <i>as-number</i> Example: Router(config-if)# ipv6 router eigrp 1	Enters router configuration mode and creates an EIGRP IPv6 routing process.
Step 9	router-id { <i>ip-address</i> <i>ipv6-address</i> } Example: Router(config-router)# router-id 10.1.1.1	Enables the use of a fixed router ID.
Step 10	no shutdown Example: Router(config-if) no shutdown	Allows the user to start the EIGRP for IPv6 protocol without changing any per-interface configuration.
Step 11	exit Example: Router(config-router) exit	Enter three times to return to privileged EXEC mode.
Step 12	show ipv6 eigrp interfaces [<i>type number</i>] [<i>as-number</i>] Example: Router# show ipv6 eigrp interfaces	Displays information about interfaces configured for EIGRP for IPv6.

Configuring the Percentage of Link Bandwidth Used by EIGRP

By default, EIGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth interface** command. You might want to change that value if a different level of link utilization is required or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

Perform the following task to configure the percentage of bandwidth that may be used by EIGRP on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **bandwidth** {*kbits* | **inherit** [*kbits*]}
5. **ipv6 bandwidth-percent eigrp** *as-number percent*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	bandwidth { <i>kbits</i> inherit [<i>kbits</i>]} Example: Router(config-if)# bandwidth 56	Sets and communicates the current bandwidth value for an interface to higher-level protocols.
Step 5	ipv6 bandwidth-percent eigrp <i>as-number percent</i> Example: Router(config-if)# ipv6 bandwidth-percent eigrp 1 75	Configures the percentage of bandwidth that may be used by EIGRP for IPv6 on an interface

Configuring Summary Aggregate Addresses

Perform this task to configure a summary aggregate address for a specified interface. If any more specific routes are in the routing table, EIGRP for IPv6 will advertise the summary address out the interface with a metric equal to the minimum of all more specific routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 summary-address eigrp** *as-number ipv6-address [admin-distance]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	ipv6 summary-address eigrp <i>as-number</i> <i>ipv6-address [admin-distance]</i> Example: Router(config-if)# ipv6 summary-address eigrp 1 2001:0DB8:0:1::/64	Configures a summary aggregate address for a specified interface.

Configuring EIGRP Route Authentication

EIGRP route authentication provides message digest algorithm 5 (MD5) authentication of routing updates from the EIGRP routing protocol. The MD5 keyed digest in each EIGRP packet prevents the introduction of unauthorized or false routing messages from unapproved sources.

Each key has its own key identifier, which is stored locally. The combination of the key identifier and the interface associated with the message uniquely identifies the authentication algorithm and MD5 authentication key in use.

You can configure multiple keys with lifetimes. Only one authentication packet is sent, regardless of how many valid keys exist. The software examines the key numbers in order from lowest to highest, and uses the first valid key it encounters. Note that the router needs to know the time.

Perform the following task to configure authentication of EIGRP routes:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 authentication mode eigrp** *as-number md5*
5. **ipv6 authentication key-chain eigrp** *as-number key-chain*
6. **exit**
7. **key chain** *name-of-chain*

8. **key** *key-id*
9. **key-string** *text*
10. **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}
11. **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	ipv6 authentication mode eigrp <i>as-number</i> md5 Example: Router(config-if)# ipv6 authentication mode eigrp 1 md5	Specifies the type of authentication used in EIGRP for IPv6 packets.
Step 5	ipv6 authentication key-chain eigrp <i>as-number</i> <i>key-chain</i> Example: Router(config-if)# ipv6 authentication key-chain eigrp 1 chain1	Enables authentication of EIGRP for IPv6 packets.
Step 6	exit Example: Router(config-if)# exit	Exits to global configuration mode.
Step 7	key chain <i>name-of-chain</i> Example: Router(config)# key chain chain1	Identifies a group of authentication keys. <ul style="list-style-type: none"> Use the name specified in Step 5.
Step 8	key <i>key-id</i> Example: Router(config-keychain)# key 1	Identifies an authentication key on a key chain.

	Command or Action	Purpose
Step 9	key-string <i>text</i> Example: Router(config-keychain-key)# key-string chain 1	Specifies the authentication string for a key.
Step 10	accept-lifetime <i>start-time</i> { infinite <i>end-time</i> duration <i>seconds</i> } Example: Router(config-keychain-key)# accept-lifetime 14:30:00 Jan 10 2006 duration 7200	Sets the time period during which the authentication key on a key chain is received as valid.
Step 11	send-lifetime <i>start-time</i> { infinite <i>end-time</i> duration <i>seconds</i> } Example: Router(config-keychain-key)# send-lifetime 15:00:00 Jan 10 2006 duration 3600	Sets the time period during which an authentication key on a key chain is valid to be sent.

Instructing the Next Hop in EIGRP

EIGRP will, by default, set the IPv6 next-hop value to be itself for routes that it is advertising, even when advertising those routes back out the same interface where it learned them. To change this default, use the following task to instruct EIGRP to use the received next-hop value when advertising these routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 next-hop-self eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	no ipv6 next-hop-self eigrp <i>as-number</i> Example: Router(config-if)# no ipv6 next-hop-self eigrp 1	Changes the default IPv6 next-hop value and instructs EIGRP to use the received next-hop value.

Adjusting the Interval Between Hello Packets in EIGRP for IPv6

Routing devices periodically send hello packets to each other to dynamically learn of other routers on their directly attached networks. This information is used to discover neighbors and to learn when neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the **bandwidth interface** command. The default hello interval remains 5 seconds for high-speed NBMA networks. Note that for the purposes of EIGRP, Frame Relay and Switched Multimegabit Data Service (SMDS) networks may or may not be considered to be NBMA. These networks are considered NBMA if the interface has not been configured to use physical multicasting; otherwise they are not considered NBMA.

You can configure the hold time on a specified interface for a particular EIGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds. For slow-speed NBMA networks, the default hold time is 180 seconds.

Perform the following task to adjust the interval between hello packets in EIGRP for IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 hello-interval eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	ipv6 hello-interval eigrp <i>as-number seconds</i> Example: Router(config)# ipv6 hello-interval eigrp 1 10	Configures the hello interval for the EIGRP for IPv6 routing process designated by an autonomous system number.

Adjusting the Hold Time in EIGRP for IPv6

On very congested and large networks, the default hold time might not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you may want to increase the hold time. Perform the following task to adjust the hold time in EIGRP for IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 hold-time eigrp** *as-number seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	ipv6 hold-time eigrp <i>as-number seconds</i> Example: Router(config)# ipv6 hold-time eigrp 1 40	Configures the hold time for a particular EIGRP for IPv6 routing process designated by the autonomous system number.

Disabling Split Horizon in EIGRP for IPv6

Split horizon controls the sending of EIGRP update and query packets. When split horizon is enabled on an interface, update and query packets are not sent for destinations for which this interface is the next hop. Controlling update and query packets in this manner reduces the possibility of routing loops.

By default, split horizon is enabled on all interfaces.

Split horizon blocks route information from being advertised by a router out of any interface from which that information originated. This behavior usually optimizes communications among multiple routing devices, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDS), situations can arise for which this behavior is not ideal. For these situations, including networks in which you have EIGRP configured, you may want to disable split horizon.

Perform the following task to disable split horizon in EIGRP for IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 split-horizon eigrp** *as-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	no ipv6 split-horizon eigrp <i>as-number</i> Example: Router(config-if)# no ipv6 split-horizon eigrp 101	Disables EIGRP for IPv6 split horizon on the specified interface.

Configuring EIGRP Stub Routing for Greater Network Stability

The EIGRP stub routing feature can help to provide greater network stability. In the event of network instability, this feature prevents EIGRP queries from being sent over limited bandwidth links to nontransit routers. Instead, distribution routers to which the stub router is connected answer the query on behalf of the stub router. This feature greatly reduces the chance of further network instability due to congested or problematic WAN links. The EIGRP stub routing feature also simplifies the configuration and maintenance of hub-and-spoke networks. When stub routing is enabled in dual-homed remote configurations, it is no longer necessary to configure filtering on remote routers to prevent those remote routers from appearing as transit paths to the hub routers.



Caution

EIGRP stub routing should be used only on stub routers. A stub router is defined as a router connected to the network core or distribution layer through which core transit traffic should not flow. A stub router should not have any EIGRP neighbors other than distribution routers.

To configure EIGRP stub routing, perform the following tasks:

- [Configuring a Router for EIGRP Stub Routing, page 14](#)
- [Verifying EIGRP Stub Routing, page 15](#)

Configuring a Router for EIGRP Stub Routing

Perform the following task to configure a remote or spoke router for EIGRP stub routing:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 router eigrp** *as-number*
5. **stub** [**receive-only** | **connected** | **static** | **summary** | **redistributed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Specifies the interface on which EIGRP is configured.
Step 4	ipv6 router eigrp <i>as-number</i> Example: Router(config-if)# ipv6 router eigrp 1	Specifies the EIGRP for IPv6 routing process to be configured.
Step 5	stub [receive-only connected static summary redistributed] Example: Router(config-router)# stub	Configures a router as a stub using EIGRP.

Verifying EIGRP Stub Routing

Perform the following task to verify that a remote router has been configured as a stub router with EIGRP.

SUMMARY STEPS

1. **enable**
2. **show ipv6 eigrp neighbors** [*interface-type* | *as-number* | **static** | **detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
	Example: Router> enable	
Step 2	show ipv6 eigrp neighbors [<i>interface-type</i> <i>as-number</i> static detail]	Displays the neighbors discovered by EIGRP for IPv6.
	Example: Router# show ipv6 eigrp neighbors detail	

Customizing an EIGRP for IPv6 Routing Process

After you have enabled EIGRP for IPv6 on a specific interface, you can configure an EIGRP for IPv6 routing process. The following optional tasks provide information on how to configure an EIGRP for IPv6 routing process to suit your needs:

- [Logging EIGRP Neighbor Adjacency Changes, page 16](#)
- [Configuring Intervals Between Neighbor Warnings, page 17](#)
- [Adjusting the EIGRP for IPv6 Metric Weights, page 18](#)

Logging EIGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are not logged. Use the following task to enable such logging.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 router eigrp** *as-number*
5. **log-neighbor-changes**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Enters the interface on which EIGRP is configured.
Step 4	ipv6 router eigrp <i>as-number</i> Example: Router(config-if)# ipv6 router eigrp 1	Specifies the EIGRP for IPv6 routing process to be configured.
Step 5	log-neighbor-changes Example: Router(config-router)# log-neighbor-changes	Enables the logging of changes in EIGRP for IPv6 neighbor adjacencies.

Configuring Intervals Between Neighbor Warnings

When neighbor warning messages occur, they are logged by default. Use the following task to configure the interval between neighbor warning messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 router eigrp** *as-number*
5. **log-neighbor-warnings** [*seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Enters the interface on which EIGRP is configured.
Step 4	ipv6 router eigrp <i>as-number</i> Example: Router(config-if)# ipv6 router eigrp 1	Specifies the EIGRP for IPv6 routing process to be configured.
Step 5	log-neighbor-warnings [<i>seconds</i>] Example: Router(config-router)# log-neighbor-warnings 300	Configures the logging intervals of EIGRP neighbor warning messages.

Adjusting the EIGRP for IPv6 Metric Weights

EIGRP for IPv6 uses the minimum bandwidth on the path to a destination network and the total delay to compute routing metrics. You can use the **metric weights** command to adjust the default behavior of EIGRP for IPv6 routing and metric computations. For example, this adjustment allows you to tune system behavior to allow for satellite transmission. EIGRP for IPv6 metric defaults have been carefully selected to provide optimal performance in most networks.

**Note**

Adjusting EIGRP metric weights can dramatically affect network performance. Because of the complexity of this task, we recommend that you do not change the default values without guidance from an experienced network designer.

By default, the EIGRP composite metric is a 32-bit quantity that is a sum of the segment delays and the lowest segment bandwidth (scaled and inverted) for a given route. For a network of homogeneous media, this metric reduces to a hop count. For a network of mixed media (FDDI, Ethernet, and serial lines running from 9600 bits per second to T1 rates), the route with the lowest metric reflects the most desirable path to a destination.

Perform the following task to adjust the EIGRP for IPv6 metric weights.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 router eigrp** *as-number*
5. **metric weights** *tos k1 k2 k3 k4 k5*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 0/0	Enters the interface on which EIGRP is configured.
Step 4	ipv6 router eigrp <i>as-number</i> Example: Router(config-if)# ipv6 router eigrp 1	Specifies the EIGRP for IPv6 routing process to be configured.
Step 5	metric weights <i>tos k1 k2 k3 k4 k5</i> Example: Router(config-router)# metric weights 0 2 0 2 0 0	Tunes EIGRP metric calculations.

Monitoring and Maintaining EIGRP

Use of **clear** and **debug** commands helps users monitor and maintain their EIGRP for IPv6 environments. The following tasks provide commands used to monitor and maintain EIGRP for IPv6:

- [Deleting Entries from EIGRP for IPv6 Routing Tables, page 19](#)
- [Using Debugging Commands to Troubleshoot an EIGRP for IPv6 Environment, page 20](#)

Deleting Entries from EIGRP for IPv6 Routing Tables

Perform the following task to delete entries from EIGRP for IPv6 routing tables.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 eigrp** [*as-number*] [**neighbor** [*ipv6-address* | *interface-type interface-number*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear ipv6 eigrp [<i>as-number</i>] [neighbor [<i>ipv6-address</i> <i>interface-type interface-number</i>]] Example: Router# clear ipv6 eigrp neighbor 3FEE:12E1:2AC1:EA32	Deletes entries from EIGRP for IPv6 routing tables.

Using Debugging Commands to Troubleshoot an EIGRP for IPv6 Environment

The use of **debug** commands can provide information users need to help troubleshoot an EIGRP for IPv6 environment. This task shows the commands used to display debugging information in EIGRP for IPv6. Note that the **debug** commands shown need not be used in any particular order and may be used only as needed.

SUMMARY STEPS

1. **enable**
2. **debug eigrp fsm**
3. **debug eigrp neighbor** [*siatimer*] [*static*]
4. **debug eigrp packet**
5. **debug eigrp transmit** [*ack*] [*build*] [*detail*] [*link*] [*packetize*] [*peerdown*] [*sia*] [*startup*] [*strange*]
6. **debug ipv6 eigrp** [*as-number*] [**neighbor** *ipv6-address* | **notification** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	debug eigrp fsm Example: Router# debug eigrp fsm	Displays debugging information about EIGRP feasible successor metrics (FSMs).
Step 3	debug eigrp neighbor [sialtimer] [static] Example: Router# debug eigrp neighbor	Displays neighbors discovered by EIGRP for IPv6.
Step 4	debug eigrp packet Example: Router# debug eigrp packet	Displays debugging information for EIGRP for IPv6 packets.
Step 5	debug eigrp transmit [ack] [build] [detail] [link] [packetize] [peerdown] [sia] [startup] [strange] Example: Router# debug eigrp transmit	Display transmittal messages sent by EIGRP for IPv6.
Step 6	debug ipv6 eigrp [as-number] [neighbor ipv6-address notification summary] Example: Router# debug ipv6 eigrp	Displays information about the EIGRP for IPv6 protocol.

Configuration Examples for Implementing EIGRP for IPv6

This section contains the following configuration example:

- [Configuring EIGRP to Establish Adjacencies on an Interface, page 21](#)

Configuring EIGRP to Establish Adjacencies on an Interface

EIGRP for IPv6 is configured directly on the interfaces over which it runs. This example shows the minimal configuration required for EIGRP for IPv6 to send hello packets in order to establish adjacencies on Ethernet 0:

```
ipv6 unicast-routing
interface e0
  ipv6 enable
  ipv6 eigrp 1
```

```

no shutdown
!
ipv6 router eigrp 1
 router-id 10.1.1.1
no shutdown

```

Where to Go Next

If you want to implement IPv6 interior gateway routing protocols, refer to the [Implementing RIP for IPv6](#) or [Implementing IS-IS for IPv6](#) module. To implement exterior gateway routing protocol Border Gateway Protocol (BGP), refer to the [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing EIGRP for IPv6 feature.

Related Documents

Related Topic	Document Title
EIGRP for IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 supported feature list	Start Here: Cisco IOS Software Release Specifics for IPv6 Features
Implementing IS-IS for IPv6	Implementing IS-IS for IPv6
Implementing Multiprotocol BGP for IPv6	Implementing Multiprotocol BGP for IPv6
Implementing RIP for IPv6	Implementing RIP for IPv6
EIGRP for IPv4	“ Configuring EIGRP ,” <i>Cisco IOS IP Routing Protocols Configuration Guide</i>
EIGRP for IPv4 commands	“ EIGRP Commands ,” <i>Cisco IOS IP Routing Protocols Command Reference</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **accept-lifetime**
- **bandwidth (interface)**
- **clear ipv6 eigrp**
- **debug eigrp fsm**
- **debug eigrp neighbor**

- **debug eigrp packet**
- **debug eigrp transmit**
- **debug ipv6 eigrp**
- **default-metric (EIGRP)**
- **distance (IPv6 EIGRP)**
- **distribute-list prefix-list (IPv6 EIGRP)**
- **ipv6 authentication key-chain eigrp**
- **ipv6 authentication mode eigrp**
- **ipv6 bandwidth-percent eigrp**
- **ipv6 eigrp**
- **ipv6 enable**
- **ipv6 hello-interval eigrp**
- **ipv6 hold-time eigrp**
- **ipv6 next-hop-self eigrp**
- **ipv6 router eigrp**
- **ipv6 split-horizon eigrp**
- **ipv6 summary-address eigrp**
- **ipv6 unicast-routing**
- **key**
- **key-chain**
- **key-string (authentication)**
- **log-neighbor-changes (IPv6 EIGRP)**
- **log-neighbor-warnings**
- **maximum-paths (IPv6)**
- **metric weights (EIGRP)**
- **neighbor (EIGRP)**
- **passive-interface (IPv6)**
- **redistribute (IPv6)**
- **router-id (IPv6)**
- **send-lifetime**
- **show ipv6 eigrp interfaces**
- **show ipv6 eigrp neighbors**
- **show ipv6 eigrp topology**
- **show ipv6 eigrp traffic**
- **stub**
- **timers active-time**
- **variance (EIGRP)**

Feature Information for Implementing EIGRP for IPv6

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.4(6)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 **Feature Information for Implementing EIGRP for IPv6**

Feature Name	Releases	Feature Information
IPv6 Routing—EIGRP Support	12.4(6)T 12.2(33)SRB 12.2(33)SXI	<p>Customers can configure EIGRP to route IPv6 prefixes. There is no linkage between EIGRP for IPv4 and EIGRP for IPv6; they are configured and managed separately. However, the configuration of EIGRP for IPv4 and IPv6 is similar and provides operational familiarity and continuity. The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • “Cisco EIGRP for IPv6 Implementation” section on page 2 • “Enabling EIGRP for IPv6 on an Interface” section on page 4

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006–2009 Cisco Systems, Inc. All rights reserved.



Configuring First Hop Redundancy Protocols in IPv6

First Published: November 28, 2006

Last Updated: November 11, 2008

IPv6 routing protocols ensure router-to-router resilience and failover. However, in situations in which the path between a host and the first-hop router fails, or the first-hop router itself fails, first hop redundancy protocols (FHRPs) ensure host-to-router resilience and failover.

The Gateway Load Balancing Protocol (GLBP) FHRP protects data traffic from a failed router or circuit, while allowing packet load sharing between a group of redundant routers. The Hot Standby Router Protocol (HSRP) protects data traffic in case of a gateway failure.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for First Hop Redundancy Protocols in IPv6”](#) section on page 29.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Prerequisites for First Hop Redundancy Protocols in IPv6, page 2](#)
- [Information About First Hop Redundancy Protocols in IPv6, page 2](#)
- [How to Configure First Hop Redundancy Protocols in IPv6, page 7](#)
- [Configuration Examples for First Hop Redundancy Protocols in IPv6, page 23](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- [Additional References, page 26](#)
- [Command Reference, page 28](#)
- [Feature Information for First Hop Redundancy Protocols in IPv6, page 29](#)

Prerequisites for First Hop Redundancy Protocols in IPv6

- Before configuring GLBP, ensure that the routers can support multiple MAC addresses on the physical interfaces. For each GLBP forwarder to be configured, an additional MAC address is used.
- Avoid static link-local addressing on interfaces configured with GLBP.
- HSRP version 2 must be enabled on an interface before HSRP IPv6 can be configured.

Information About First Hop Redundancy Protocols in IPv6

Before you configure any FHRPs in IPv6, you should understand the following concepts:

- [GLBP for IPv6, page 2](#)
- [HSRP for IPv6, page 6](#)

GLBP for IPv6

This section provides the following information about GLBP for IPv6:

- [GLBP for IPv6 Overview, page 2](#)
- [GLBP Benefits, page 3](#)
- [GLBP Active Virtual Gateway, page 3](#)
- [GLBP Virtual MAC Address Assignment, page 4](#)
- [GLBP Virtual Gateway Redundancy, page 4](#)
- [GLBP Virtual Forwarder Redundancy, page 5](#)
- [GLBP Gateway Priority, page 5](#)
- [GLBP Gateway Weighting and Tracking, page 5](#)

GLBP for IPv6 Overview

The Gateway Load Balancing Protocol feature provides automatic router backup for IPv6 hosts configured with a single default gateway on an IEEE 802.3 LAN. Multiple first hop routers on the LAN combine to offer a single virtual first-hop IPv6 router while sharing the IPv6 packet forwarding load. GLBP performs a similar function for the user as HSRP. HSRP allows multiple routers to participate in a virtual router group configured with a virtual IPv6 address. One member is elected to be the active router to forward packets sent to the virtual IPv6 address for the group. The other routers in the group are redundant until the active router fails. These standby routers have unused bandwidth that the protocol is not using. Although multiple virtual router groups can be configured for the same set of routers, the hosts must be configured for different default gateways, which results in an extra administrative burden. The advantage of GLBP is that it additionally provides load balancing over multiple routers (gateways) using a single virtual IPv6 address and multiple virtual MAC addresses. The forwarding load is shared

among all routers in a GLBP group rather than being handled by a single router while the other routers stand idle. Each host is configured with the same virtual IPv6 address, and all routers in the virtual router group participate in forwarding packets.

GLBP Benefits

GLBP for IPv6 provides the following benefits:

Load Sharing

You can configure GLBP in such a way that traffic from LAN clients can be shared equitably among multiple routers.

Multiple Virtual Routers

GLBP supports up to 1024 virtual routers (GLBP groups) on each physical interface of a router and up to four virtual forwarders per group.

Preemption

The redundancy scheme of GLBP enables you to preempt an active virtual gateway with a higher priority backup virtual gateway that has become available. Forwarder preemption works in a similar way, except that forwarder preemption uses weighting instead of priority and is enabled by default.

Authentication

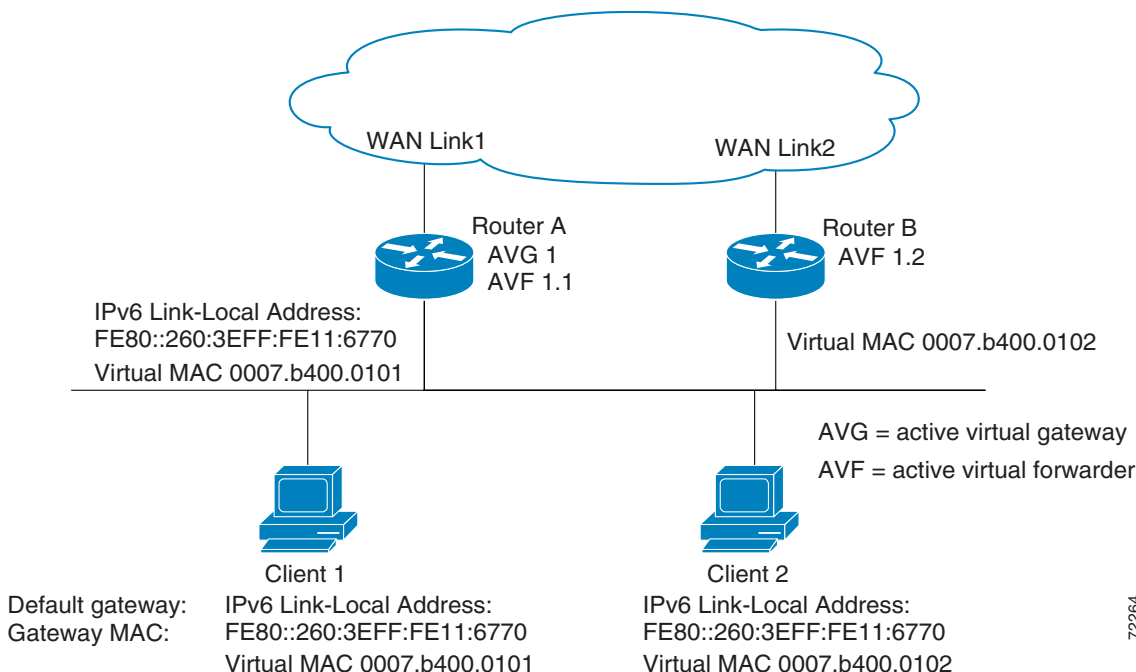
You can also use the industry-standard Message Digest algorithm 5 (MD5) algorithm for improved reliability, security, and protection against GLBP-spoofing software. A router within a GLBP group with a different authentication string than other routers will be ignored by other group members. You can alternatively use a simple text password authentication scheme between GLBP group members to detect configuration errors.

GLBP Active Virtual Gateway

Members of a GLBP group elect one gateway to be the active virtual gateway (AVG) for that group. Other group members provide backup for the AVG in the event that the AVG becomes unavailable. The function of the AVG is that it assigns a virtual MAC address to each member of the GLBP group. Each gateway assumes responsibility for forwarding packets sent to the virtual MAC address assigned to it by the AVG. These gateways are known as active virtual forwarders (AVFs) for their virtual MAC address.

The AVG is also responsible for answering Address Resolution Protocol (ARP) requests for the virtual IPv6 address. Load sharing is achieved by the AVG replying to the ARP requests with different virtual MAC addresses.

In [Figure 1](#), Router A is the AVG for a GLBP group, and is responsible for the IPv6 link-local address FE80::260:3EFF:FE11:6770. Router A is also an AVF for the virtual MAC address 0007.b400.0101. Router B is a member of the same GLBP group and is designated as the AVF for the virtual MAC address 0007.b400.0102. Client 1 has a default gateway IPv6 address of FE80::260:3EFF:FE11:6770 and a gateway MAC address of 0007.b400.0101. Client 2 shares the same default gateway IP address but receives the gateway MAC address 0007.b400.0102 because Router B is sharing the traffic load with Router A.

Figure 1 **GLBP Topology**

If Router A becomes unavailable, Client 1 will not lose access to the WAN because Router B will assume responsibility for forwarding packets sent to the virtual MAC address of Router A, and for responding to packets sent to its own virtual MAC address. Router B will also assume the role of the AVG for the entire GLBP group. Communication for the GLBP members continues despite the failure of a router in the GLBP group.

GLBP Virtual MAC Address Assignment

A GLBP group allows up to four virtual MAC addresses per group. The AVG is responsible for assigning the virtual MAC addresses to each member of the group. Other group members request a virtual MAC address after they discover the AVG through hello messages. Gateways are assigned the next MAC address in sequence. A virtual forwarder that is assigned a virtual MAC address by the AVG is known as a primary virtual forwarder. Other members of the GLBP group learn the virtual MAC addresses from hello messages. A virtual forwarder that has learned the virtual MAC address is referred to as a secondary virtual forwarder.

GLBP Virtual Gateway Redundancy

GLBP operates virtual gateway redundancy in the same way as HSRP. One gateway is elected as the AVG, another gateway is elected as the standby virtual gateway, and the remaining gateways are placed in a listen state.

If an AVG fails, the standby virtual gateway will assume responsibility for the virtual IPv6 address. A new standby virtual gateway is then elected from the gateways in the listen state.

GLBP Virtual Forwarder Redundancy

GLBP virtual forwarder redundancy is similar to virtual gateway redundancy with an AVF. If the AVF fails, one of the secondary virtual forwarders in the listen state assumes responsibility for the virtual MAC address.

The new AVF is also a primary virtual forwarder for a different forwarder number. GLBP migrates hosts away from the old forwarder number using two timers that start as soon as the gateway changes to the active virtual forwarder state. GLBP uses the hello messages to communicate the current state of the timers.

The redirect time is the interval during which the AVG continues to redirect hosts to the old virtual forwarder MAC address. When the redirect time expires, the AVG stops using the old virtual forwarder MAC address in ARP replies, although the virtual forwarder will continue to forward packets that were sent to the old virtual forwarder MAC address.

The secondary hold time is the interval during which the virtual forwarder is valid. When the secondary hold time expires, the virtual forwarder is removed from all gateways in the GLBP group. The expired virtual forwarder number becomes eligible for reassignment by the AVG.

GLBP Gateway Priority

GLBP gateway priority determines the role that each GLBP gateway plays and the results if the AVG fails.

Priority also determines if a GLBP router functions as a backup virtual gateway and the order of ascendancy to becoming an AVG if the current AVG fails. You can configure the priority of each backup virtual gateway with a value of 1 through 255 using the **glbp priority** command.

In [Figure 1](#), if Router A—the AVG in a LAN topology—fails, an election process takes place to determine which backup virtual gateway should take over. In this example, Router B is the only other member in the group so it will automatically become the new AVG. If another router existed in the same GLBP group with a higher priority, then the router with the higher priority would be elected. If both routers have the same priority, the backup virtual gateway with the higher IPv6 address would be elected to become the active virtual gateway.

By default, the GLBP virtual gateway preemptive scheme is disabled. A backup virtual gateway can become the AVG only if the current AVG fails, regardless of the priorities assigned to the virtual gateways. You can enable the GLBP virtual gateway preemptive scheme using the **glbp preempt** command. Preemption allows a backup virtual gateway to become the AVG, if the backup virtual gateway is assigned a higher priority than the current AVG.

GLBP Gateway Weighting and Tracking

GLBP uses a weighting scheme to determine the forwarding capacity of each router in the GLBP group. The weighting assigned to a router in the GLBP group can be used to determine whether it will forward packets and, if so, the proportion of hosts in the LAN for which it will forward packets. Thresholds can be set to disable forwarding when the weighting falls below a certain value. When the weighting rises above another threshold, forwarding is automatically reenabled.

The GLBP group weighting can be automatically adjusted by tracking the state of an interface within the router. If a tracked interface goes down, the GLBP group weighting is reduced by a specified value. Different interfaces can be tracked to decrement the GLBP weighting by varying amounts.

By default, the GLBP virtual forwarder preemptive scheme is enabled with a delay of 30 seconds. A backup virtual forwarder can become the AVF if the current AVF weighting falls below the low weighting threshold for 30 seconds. You can disable the GLBP forwarder preemptive scheme using the **no glbp forwarder preempt** command or change the delay using the **glbp forwarder preempt delay minimum** command.

HSRP for IPv6

This section provides the following information about GLBP for IPv6:

- [HSRP for IPv6 Overview, page 6](#)
- [HSRP IPv6 Virtual MAC Address Range, page 6](#)
- [HSRP IPv6 UDP Port Number, page 6](#)

HSRP for IPv6 Overview

The HSRP is an FHRP designed to allow for transparent failover of the first-hop IP router. HSRP provides high network availability by providing first-hop routing redundancy for IP hosts on Ethernet configured with a default gateway IP address. HSRP is used in a group of routers for selecting an active router and a standby router. In a group of router interfaces, the active router is the router of choice for routing packets; the standby router is the router that takes over when the active router fails or when preset conditions are met.

IPv6 hosts learn of available IPv6 routers through IPv6 neighbor discovery RA messages. These are multicast periodically, or may be solicited by hosts. HSRP is designed to provide only a virtual first hop for IPv6 hosts.

An HSRP IPv6 group has a virtual MAC address that is derived from the HSRP group number, and a virtual IPv6 link-local address that is, by default, derived from the HSRP virtual MAC address. Periodic RAs are sent for the HSRP virtual IPv6 link-local address when the HSRP group is active. These RAs stop after a final RA is sent when the group leaves the active state.

Periodic RAs for the interface link-local address stop after a final RA is sent while at least one virtual IPv6 link-local address is configured on the interface. No restrictions occur for the interface IPv6 link-local address other than that mentioned for the RAs. Other protocols continue to receive and send packets to this address.

HSRP uses a priority mechanism to determine which HSRP configured router is to be the default active router. To configure a router as the active router, you assign it a priority that is higher than the priority of all the other HSRP-configured routers. The default priority is 100, so if you configure just one router to have a higher priority, that router will be the default active router.

HSRP IPv6 Virtual MAC Address Range

HSRP IPv6 uses a different virtual MAC address block than does HSRP for IP:

0005.73A0.0000 through 0005.73A0.0FFF (4096 addresses)

HSRP IPv6 UDP Port Number

Port number 2029 has been assigned to HSRP IPv6.

How to Configure First Hop Redundancy Protocols in IPv6

This section describes how to configure FHRPs.

- [Configuring and Customizing GLBP, page 7](#)
- [Enabling an HSRP Group for IPv6 Operation, page 20](#)

Configuring and Customizing GLBP

Customizing the behavior of GLBP is optional. Be aware that as soon as you enable a GLBP group, that group is operating. It is possible that if you first enable a GLBP group before customizing GLBP, the router could take over control of the group and become the AVG before you have finished customizing the feature. Therefore, if you plan to customize GLBP, it is a good idea to do so before enabling GLBP.

This section contains the following procedures:

- [Customizing GLBP, page 7](#) (optional)
- [Configuring GLBP Authentication, page 9](#) (optional)
- [Configuring GLBP Weighting Values and Object Tracking, page 15](#) (optional)
- [Troubleshooting the GLBP, page 18](#) (required)

Customizing GLBP

This task explains how to customize your GLBP configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** { *ipv6-address/prefix-length* | *prefix-name ipv6-prefix/prefix-length* | **autoconfig** [*default-route*] }
5. **glbp group timers** [*msec*] *hellotime* [*msec*] *holdtime*
6. **glbp group timers redirect** *redirect timeout*
7. **glbp group load-balancing** [*host-dependent* | **round-robin** | **weighted**]
8. **glbp group priority** *level*
9. **glbp group preempt** [*delay minimum seconds*]
10. **glbp group name** *redundancy-name*
11. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface fastethernet 0/0	Specifies an interface type and number, and enters interface configuration mode.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name ipv6-prefix/prefix-length</i> autoconfig [<i>default-route</i>]} Example: Router(config-if)# ipv6 address 2001:0DB8:0:7272::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	glbp group timers [<i>msec</i>] <i>hellotime</i> [<i>msec</i>] <i>holdtime</i> Example: Router(config-if)# glbp 10 timers 5 18	Configures the interval between successive hello packets sent by the AVG in a GLBP group. <ul style="list-style-type: none"> The <i>holdtime</i> argument specifies the interval in seconds before the virtual gateway and virtual forwarder information in the hello packet is considered invalid. The optional msec keyword specifies that the following argument will be expressed in milliseconds, instead of the default seconds.
Step 6	glbp group timers redirect <i>redirect timeout</i> Example: Router(config-if)# glbp 10 timers redirect 600 7200	Configures the time interval during which the AVG continues to redirect clients to an AVF. <ul style="list-style-type: none"> The <i>timeout</i> argument specifies the interval in seconds before a secondary virtual forwarder becomes invalid.
Step 7	glbp group load-balancing [host-dependent round-robin weighted] Example: Router(config-if)# glbp 10 load-balancing host-dependent	Specifies the method of load balancing used by the GLBP AVG.
Step 8	glbp group priority <i>level</i> Example: Router(config-if)# glbp 10 priority 254	Sets the priority level of the gateway within a GLBP group. <ul style="list-style-type: none"> The default value is 100.

	Command or Action	Purpose
Step 9	glbp group preempt [delay minimum seconds] Example: Router(config-if)# glbp 10 preempt delay minimum 60	Configures the router to take over as AVG for a GLBP group if it has a higher priority than the current AVG. <ul style="list-style-type: none"> This command is disabled by default. Use the optional delay and minimum keywords and the <i>seconds</i> argument to specify a minimum delay interval in seconds before preemption of the AVG takes place.
Step 10	glbp group name <i>redundancy-name</i> Example: Router(config-if)# glbp 10 name abcompany	Enables IPv6 redundancy by assigning a name to the GLBP group. <ul style="list-style-type: none"> The GLBP redundancy client must be configured with the same GLBP group name so the redundancy client and the GLBP group can be connected.
Step 11	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.

Configuring GLBP Authentication

The following sections describe configuration tasks for GLBP authentication. The task you perform depends on whether you want to use text authentication, a simple MD5 key string, or MD5 key chains for authentication.

- [Configuring GLBP MD5 Authentication Using a Key String, page 10](#)
- [Configuring GLBP MD5 Authentication Using a Key Chain, page 11](#)
- [Configuring GLBP Text Authentication, page 13](#)

How GLBP MD5 Authentication Works

GLBP MD5 authentication provides greater security than the alternative plain text authentication scheme. MD5 authentication allows each GLBP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. A keyed hash of an incoming packet is generated and, if the hash within the incoming packet does not match the generated hash, the packet is ignored.

The key for the MD5 hash can either be given directly in the configuration using a key string or supplied indirectly through a key chain.

A router will ignore incoming GLBP packets from routers that do not have the same authentication configuration for a GLBP group. GLBP has three authentication schemes:

- No authentication
- Plain text authentication
- MD5 authentication

GLBP packets will be rejected in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- MD5 digests differ on the router and in the incoming packet.
- Text authentication strings differ on the router and in the incoming packet.

Configuring GLBP MD5 Authentication Using a Key String

Configuring GLBP MD5 authentication protects the router against spoofing software and uses the industry-standard MD5 algorithm for improved reliability and security. Perform this task to configure GLBP MD5 authentication using a key string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** { *ipv6-address/prefix-length* | *prefix-name ipv6-prefix/prefix-length* | **autoconfig** [*default-route*] }
5. **glbp group-number authentication md5 key-string** [0 | 7] *key*
6. **glbp group ipv6** [*ipv6-address* | **autoconfig**]
7. Repeat Steps 1 through 6 on each router that will communicate.
8. **end**
9. **show glbp**

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/1	Configures an interface type and enters interface configuration mode.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name ipv6-prefix/prefix-length</i> autoconfig [<i>default-route</i>] } Example: Router(config-if)# ipv6 address 2001:0DB8:0:7272::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.

	Command	Purpose
Step 5	glbp group-number authentication md5 key-string [0 7] key Example: Router(config-if)# glbp 1 authentication md5 key-string d00b4r987654321a	Configures an authentication key for GLBP MD5 authentication. <ul style="list-style-type: none"> The number of characters in the command plus the key string must not exceed 255 characters. No keyword before the <i>key</i> argument or specifying 0 means the key is unencrypted. Specifying 7 means the key is encrypted. The key-string authentication key will automatically be encrypted if the service password-encryption global configuration command is enabled.
Step 6	glbp group ipv6 [ipv6-address autoconfig] Example: Router(config-if)# glbp 1 ipv6 FE80::260:3EFF:FE11:6770	Enables GLBP in IPv6.
Step 7	Repeat Steps 1 through 6 on each router that will communicate.	—
Step 8	end Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 9	show glbp Example: Router# show glbp	(Optional) Displays GLBP information. <ul style="list-style-type: none"> Use this command to verify your configuration. The key string and authentication type will be displayed if configured.

Configuring GLBP MD5 Authentication Using a Key Chain

Perform this task to configure GLBP MD5 authentication using a key chain. Key chains allow a different key string to be used at different times according to the key chain configuration. GLBP will query the appropriate key chain to obtain the current live key and key ID for the specified key chain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain name-of-chain**
4. **key key-id**
5. **key-string string**
6. **exit**
7. **exit**
8. **interface type number**

9. **ipv6 address** { *ipv6-address/prefix-length* | *prefix-name ipv6-prefix/prefix-length* | **autoconfig** [*default-route*]
10. **glbp group-number authentication md5 key-chain** *name-of-chain*
11. **glbp group ipv6** [*ipv6-address* | **autoconfig**]
12. Repeat Steps 1 through 11 on each router that will communicate.
13. **end**
14. **show glbp**
15. **show key chain**

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	key chain <i>name-of-chain</i> Example: Router(config)# key chain glbp2	Enables authentication for routing protocols and identifies a group of authentication keys.
Step 4	key <i>key-id</i> Example: Router(config-keychain)# key 100	Identifies an authentication key on a key chain. <ul style="list-style-type: none">The <i>key-id</i> must be a number.
Step 5	key-string <i>string</i> Example: Router(config-keychain-key)# key-string string1	Specifies the authentication string for a key. <ul style="list-style-type: none">The <i>string</i> can be 1 to 80 uppercase or lowercase alphanumeric characters; the first character cannot be a numeral.
Step 6	exit Example: Router(config-keychain-key)# exit	Returns to keychain configuration mode.
Step 7	exit Example: Router(config-keychain)# exit	Returns to global configuration mode.

	Command	Purpose
Step 8	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/1	Configures an interface type and enters interface configuration mode.
Step 9	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name</i> <i>ipv6-prefix/prefix-length</i> autoconfig [<i>default-route</i>]} Example: Router(config-if)# ipv6 address 2001:0DB8:0:7272::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 10	glbp <i>group-number</i> authentication md5 key-chain <i>name-of-chain</i> Example: Router(config-if)# glbp 1 authentication md5 key-chain glbp2	Configures an authentication MD5 key chain for GLBP MD5 authentication. <ul style="list-style-type: none">The key chain name must match the name specified in Step 3.
Step 11	glbp <i>group</i> ipv6 [<i>ipv6-address</i> autoconfig] Example: Router(config-if)# glbp 1 ipv6 FE80::E0:F727:E400:A	Enables GLBP in IPv6.
Step 12	Repeat Steps 1 through 11 on each router that will communicate.	—
Step 13	end Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 14	show glbp Example: Router# show glbp	(Optional) Displays GLBP information. <ul style="list-style-type: none">Use this command to verify your configuration. The key chain and authentication type will be displayed if configured.
Step 15	show key chain Example: Router# show key chain	(Optional) Displays authentication key information.

Configuring GLBP Text Authentication

Perform this task to configure GLBP text authentication. This method of authentication provides minimal security. Use MD5 authentication if security is required.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface** *type number*
4. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name ipv6-prefix/prefix-length* | **autoconfig** [*default-route*]}
5. **glbp group-number authentication text string**
6. **glbp group ipv6** [*ipv6-address* | **autoconfig**]
7. Repeat Steps 1 through 6 on each router that will communicate.
8. **end**
9. **show glbp**

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/1	Configures an interface type and enters interface configuration mode.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name ipv6-prefix/prefix-length</i> autoconfig [<i>default-route</i>]} Example: Router(config-if)# ipv6 address 2001:0DB8:0:7272::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	glbp group-number authentication text string Example: Router(config-if)# glbp 10 authentication text stringxyz	Authenticates GLBP packets received from other routers in the group. <ul style="list-style-type: none">• If you configure authentication, all routers within the GLBP group must use the same authentication string.
Step 6	glbp group ipv6 [<i>ipv6-address</i> autoconfig] Example: Router(config-if)# glbp 1 ipv6 FE80::60:3E47:AC8:8	Enables GLBP in IPv6.
Step 7	Repeat Steps 1 through 6 on each router that will communicate.	—

	Command	Purpose
Step 8	end Example: Router(config-if)# end	Returns to privileged EXEC mode.
Step 9	show glbp Example: Router# show glbp	(Optional) Displays GLBP information. <ul style="list-style-type: none"> Use this command to verify your configuration.

Configuring GLBP Weighting Values and Object Tracking

GLBP weighting is used to determine whether a router can act as a virtual forwarder. Initial weighting values can be set and optional thresholds specified. Interface states can be tracked and a decrement value set to reduce the weighting value if the interface goes down. When the GLBP router weighting drops below a specified value, the router will no longer be an active virtual forwarder. When the weighting rises above a specified value, the router can resume its role as an active virtual forwarder.

Perform this task to configure GLBP weighting values and object tracking.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* {**line-protocol** | **ip routing**}
4. **exit**
5. **interface** *type number*
6. **glbp group weighting** *maximum* [**lower** *lower*] [**upper** *upper*]
7. **glbp group weighting track** *object-number* [**decrement** *value*]
8. **glbp group forwarder preempt** [**delay** *minimum seconds*]
9. **end**
10. **show track** [*object-number* | **brief**] [**interface** [**brief**] | **ip route** [**brief**] | **resolution** | **timers**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	<p>track <i>object-number</i> interface <i>type number</i> {line-protocol ip routing}</p> <p>Example: Router(config)# track 2 interface POS 6/0 ip routing</p>	<p>Configures an interface to be tracked where changes in the state of the interface affect the weighting of a GLBP gateway, and enters tracking configuration mode.</p> <ul style="list-style-type: none"> This command configures the interface and corresponding object number to be used with the glbp weighting track command. The line-protocol keyword tracks whether the interface is up. The ip routing keywords also check that IPv6 routing is enabled on the interface and an IPv6 address is configured.
Step 4	<p>exit</p> <p>Example: Router(config-track)# exit</p>	Returns to global configuration mode.
Step 5	<p>interface <i>type number</i></p> <p>Example: Router(config)# interface fastethernet 0/0</p>	Enters interface configuration mode.
Step 6	<p>glbp group weighting <i>maximum</i> [lower <i>lower</i>] [upper <i>upper</i>]</p> <p>Example: Router(config-if)# glbp 10 weighting 110 lower 95 upper 105</p>	Specifies the initial weighting value, and the upper and lower thresholds, for a GLBP gateway.
Step 7	<p>glbp group weighting track <i>object-number</i> [decrement <i>value</i>]</p> <p>Example: Router(config-if)# glbp 10 weighting track 2 decrement 5</p>	<p>Specifies an object to be tracked that affects the weighting of a GLBP gateway.</p> <ul style="list-style-type: none"> The <i>value</i> argument specifies a reduction in the weighting of a GLBP gateway when a tracked object fails.
Step 8	<p>glbp group forwarder preempt [delay <i>minimum seconds</i>]</p> <p>Example: Router(config-if)# glbp 10 forwarder preempt delay minimum 60</p>	<p>Configures the router to take over as the AVF for a GLBP group if the current AVF for a GLBP group falls below its low weighting threshold.</p> <ul style="list-style-type: none"> This command is enabled by default with a delay of 30 seconds. Use the optional delay and minimum keywords and the <i>seconds</i> argument to specify a minimum delay interval in seconds before preemption of the AVF takes place.

	Command or Action	Purpose
Step 9	end Example: Router(config-if)# exit	Returns to privileged EXEC mode.
Step 10	show track [<i>object-number</i> brief] [interface [brief] ip route [brief] resolution timers] Example: Router# show track 2	Displays tracking information.

Enabling and Verifying GLBP

This task explains how to enable GLBP on an interface and verify its configuration and operation. GLBP is designed to be easy to configure. Each gateway in a GLBP group must be configured with the same group number, and at least one gateway in the GLBP group must be configured with the virtual IPv6 address to be used by the group. All other required parameters can be learned.

Prerequisites

If VLANs are in use on an interface, the GLBP group number must be different for each VLAN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name ipv6-prefix/prefix-length* | **autoconfig** [*default-route*]}
5. **glbp group ipv6** [*ipv6-address* | **autoconfig**]
6. **exit**
7. **show glbp** [*interface-type interface-number*] [*group*] [*state*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface fastethernet 0/0	Specifies an interface type and number, and enters interface configuration mode.
Step 4	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name ipv6-prefix/prefix-length</i> autoconfig [<i>default-route</i>] } Example: Router(config-if)# ipv6 address 2001:0DB8:0:7262::62/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 5	glbp group ipv6 [<i>ipv6-address</i> autoconfig] Example: Router(config-if)# glbp 1 ipv6 FE80::60:3E47:AC8:8	Enables GLBP in IPv6.
Step 6	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 7	show glbp [<i>interface-type interface-number</i>] [<i>group</i>] [<i>state</i>] [brief] Example: Router(config)# show glbp 10	(Optional) Displays information about GLBP groups on a router. <ul style="list-style-type: none"> Use the optional brief keyword to display a single line of information about each virtual gateway or virtual forwarder.

Troubleshooting the GLBP

The Gateway Load Balancing Protocol feature introduces five privileged EXEC mode commands to enable diagnostic output concerning various events relating to the operation of GLBP to be displayed on a console. The **debug condition glbp**, **debug glbp errors**, **debug glbp events**, **debug glbp packets**, and **debug glbp terse** commands are intended only for troubleshooting purposes because the volume of output generated by the software can result in severe performance degradation on the router. Perform this task to minimize the impact of using the **debug glbp** commands.

This procedure will minimize the load on the router created by the **debug condition glbp** or **debug glbp** command because the console port will no longer generate character-by-character processor interrupts. If you cannot connect to a console directly, you can run this procedure via a terminal server. If you must break the Telnet connection, however, you may not be able to reconnect because the router may be unable to respond due to the processor load of generating the debugging output.

Prerequisites

This task requires a router running GLBP to be attached directly to a console.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no logging console**
4. Use Telnet to access a router port and repeat Steps 1 and 2.
5. **end**
6. **terminal monitor**
7. **debug condition glbp** *interface-type interface-number group [forwarder]*
8. **terminal no monitor**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	no logging console Example: Router(config)# no logging console	Disables all logging to the console terminal. <ul style="list-style-type: none">• To reenabling logging to the console, use the logging console command in global configuration mode.
Step 4	Use Telnet to access a router port and repeat Steps 1 and 2.	Enters global configuration mode in a recursive Telnet session, which allows the output to be redirected away from the console port.
Step 5	end Example: Router(config)# end	Exits to privileged EXEC mode.
Step 6	terminal monitor Example: Router# terminal monitor	Enables logging output on the virtual terminal.

	Command or Action	Purpose
Step 7	debug condition glbp <i>interface-type</i> <i>interface-number group [forwarder]</i> Example: Router# debug condition glbp fastethernet 0/0 10 1	Displays debugging messages about GLBP conditions. <ul style="list-style-type: none"> Try to enter only specific debug condition glbp or debug glbp commands to isolate the output to a certain subcomponent and minimize the load on the processor. Use appropriate arguments and keywords to generate more detailed debug information on specified subcomponents. Enter the specific no debug condition glbp or no debug glbp command when you are finished.
Step 8	terminal no monitor Example: Router# terminal no monitor	Disables logging on the virtual terminal.

Enabling an HSRP Group for IPv6 Operation

If an IPv6 address is entered, it must be link local. There are no HSRP IPv6 secondary addresses.

Prerequisites

HSRP version 2 must be enabled on an interface before HSRP IPv6 can be configured.

The following tasks describe how to enable and verify an HSRP group for IPv6:

- [Enabling HSRP Version 2, page 20](#)
- [Enabling and Verifying an HSRP Group for IPv6 Operation, page 21](#)

Enabling HSRP Version 2

The following task describes how to enable HSRP version 2 on an interface before HSRP IPv6 can be configured.

SUMMARY STEPS

- enable**
- configure terminal**
- interface** *type number*
- standby version** {1 | 2}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	standby version {1 2} Example: Router(config-if)# standby version 2	Changes the version of the HSRP. <ul style="list-style-type: none"> Version 1 is the default.

Enabling and Verifying an HSRP Group for IPv6 Operation

In this task, when you enter the **standby ipv6** command, a link-local address is generated from the link-local prefix, and a modified EUI-64 format interface identifier is generated in which the EUI-64 interface identifier is created from the relevant HSRP virtual MAC address.

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need site-local or globally unique addresses to communicate.

In IPv6, a router on the link advertises in RA messages any site-local and global prefixes, and its willingness to function as a default router for the link. RA messages are sent periodically and in response to router solicitation messages, which are sent by hosts at system startup.

A node on the link can automatically configure site-local and global IPv6 addresses by appending its interface identifier (64 bits) to the prefixes (64 bits) included in the RA messages. The resulting 128-bit IPv6 addresses configured by the node are then subjected to duplicate address detection to ensure their uniqueness on the link. If the prefixes advertised in the RA messages are globally unique, then the IPv6 addresses configured by the node are also guaranteed to be globally unique. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message.

The following task describes how to enable an HSRP group for IPv6 operation and verify HSRP information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **standby** [*group-number*] **ipv6** {*link-local-address* | **autoconfig**}
6. **standby** [*group-number*] **preempt** [**delay** {*minimum seconds* | **reload** *seconds* | **sync** *seconds*}]
7. **standby** [*group-number*] **priority** *priority*
8. **exit**
9. **show standby** [*type number* [*group*]] [**all** | **brief**]
10. **show ipv6 interface** [**brief**] [*interface-type interface-number*] [**prefix**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams. <ul style="list-style-type: none"> The ipv6 unicast-routing command must be enabled for HSRP for IPv6 to work.
Step 4	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	standby [<i>group-number</i>] ipv6 { <i>link-local-address</i> autoconfig }	Activates the HSRP in IPv6.
Step 6	standby [<i>group-number</i>] preempt [delay { <i>minimum seconds</i> reload <i>seconds</i> sync <i>seconds</i> }] Example: Router(config-if)# standby 1 preempt	Configures HSRP preemption and preemption delay.

	Command or Action	Purpose
Step 7	standby [<i>group-number</i>] priority <i>priority</i> Example: Router(config-if)# standby 1 priority 110	Configures HSRP priority.
Step 8	exit Example: Router(config-if)# exit	Returns the router to privileged EXEC mode.
Step 9	show standby [<i>type number</i> [<i>group</i>]] [all brief] Example: Router# show standby	Displays HSRP information.
Step 10	show ipv6 interface [brief] [<i>interface-type</i> <i>interface-number</i>] [prefix] Example: Router# show ipv6 interface ethernet 0/0	Displays the usability status of interfaces configured for IPv6.

Configuration Examples for First Hop Redundancy Protocols in IPv6

This section has the following configuration examples:

- [Customizing GLBP Configuration: Example, page 23](#)
- [GLBP MD5 Authentication Using Key Strings: Example, page 24](#)
- [GLBP MD5 Authentication Using Key Chains: Example, page 24](#)
- [GLBP Text Authentication: Example, page 24](#)
- [GLBP Weighting: Example, page 24](#)
- [Enabling GLBP Configuration: Example, page 24](#)
- [Enabling and Verifying an HSRP Group for IPv6 Operation: Example, page 25](#)

Customizing GLBP Configuration: Example

In the following example, Router A, shown in Figure 1, is configured with a number of GLBP commands:

```
interface fastethernet 0/0
 ipv6 address 2001:0DB8:0001:0001::/64
 glbp 10 timers 5 18
 glbp 10 timers redirect 600 7200
 glbp 10 load-balancing host-dependent
 glbp 10 priority 254
 glbp 10 preempt delay minimum 60
```

GLBP MD5 Authentication Using Key Strings: Example

The following example configures GLBP MD5 authentication using a key string:

```
!
interface Ethernet 0/1
 ipv6 address 2001:0DB8:0001:0001:/64
 glbp 2 authentication md5 key-string ThisStringIsTheSecretKey
 glbp 2 ipv6 FE80::260:3EFF:FE11:6770
```

GLBP MD5 Authentication Using Key Chains: Example

In the following example, GLBP queries the key chain “AuthenticateGLBP” to obtain the current live key and key ID for the specified key chain:

```
key chain AuthenticateGLBP
 key 1
  key-string ThisIsASecretKey
interface Ethernet 0/1
 ipv6 address 2001:0DB8:0001:0001:/64
 glbp 2 authentication md5 key-chain AuthenticateGLBP
 glbp 2 ipv6 FE80::E0:F727:E400:A
```

GLBP Text Authentication: Example

The following example configures GLBP text authentication using a text string:

```
interface fastethernet 0/0
 ipv6 address 2001:0DB8:0001:0001:/64
 glbp 10 authentication text stringxyz
 glbp 10 ipv6 FE80::60:3E47:AC8:8
```

GLBP Weighting: Example

In the following example, Router A, shown in Figure 1, is configured to track the IP routing state of the POS interfaces 5/0 and 6/0, an initial GLBP weighting with upper and lower thresholds is set, and a weighting decrement value of 10 is set. If POS interfaces 5/0 and 6/0 go down, the weighting value of the router is reduced.

```
track 1 interface POS 5/0 ip routing
track 2 interface POS 6/0 ip routing
interface fastethernet 0/0
 glbp 10 weighting 110 lower 95 upper 105
 glbp 10 weighting track 1 decrement 10
 glbp 10 weighting track 2 decrement 10
 glbp 10 forwarder preempt delay minimum 60
```

Enabling GLBP Configuration: Example

In the following example, the router is configured to enable GLBP, and the virtual IPv6 address of 2001:0DB8:0002:0002:/64 is specified for GLBP group 10:

```
interface fastethernet 0/0
 ipv6 address 2001:0DB8:0001:0001:/64
 glbp 10 ipv6 FE80::60:3E47:AC8:8
```

In the following example, GLBP for IPv6 is enabled for GLBP group 15:

```
interface fastethernet 0/0
  ipv6 address 2001:0DB8:0001:0001::/64
  glbp 10 ipv6
```

Enabling and Verifying an HSRP Group for IPv6 Operation: Example

The following example shows configuration verification for an HSRP group for IPv6 that consists of Router1 and Router2. The **show standby** command is issued for each router to verify the router's configuration.

Router 1 Configuration

```
interface FastEthernet0/0.100
description DATA VLAN for PCs
encapsulation dot1Q 100
ipv6 address 2001:DB8:CAFE:2100::BAD1:1010/64
standby version 2
standby 101 priority 120
standby 101 preempt delay minimum 30
standby 101 authentication ese
standby 101 track Serial0/1/0.17 90
standby 201 ipv6 autoconfig
standby 201 priority 120
standby 201 preempt delay minimum 30
standby 201 authentication ese
standby 201 track Serial0/1/0.17 90
```

Router1# **show standby**

```
FastEthernet0/0.100 - Group 101 (version 2)
State is Active
2 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.296 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
Active router is local
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Active
2 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 2.428 secs
Authentication text "ese"
Preemption enabled, delay min 30 secs
Active router is local
Standby router is FE80::20F:8FFF:FE37:3B70, priority 100 (expires in 7.856 sec)
Priority 120 (configured 120)
Track interface Serial0/1/0.17 state Up decrement 90
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Router 2 Configuration

```
interface FastEthernet0/0.100
description DATA VLAN for Computers
encapsulation dot1Q 100
ipv6 address 2001:DB8:CAFE:2100::BAD1:1020/64
standby version 2
standby 101 preempt
standby 101 authentication ese
standby 201 ipv6 autoconfig
standby 201 preempt
standby 201 authentication ese
```

Router2# **show standby**

```
FastEthernet0/0.100 - Group 101 (version 2)
State is Standby
7 state changes, last state change 5w5d
Active virtual MAC address is 0000.0c9f.f065
Local virtual MAC address is 0000.0c9f.f065 (v2 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-101" (default)
FastEthernet0/0.100 - Group 201 (version 2)
State is Standby
7 state changes, last state change 5w5d
Virtual IP address is FE80::5:73FF:FEA0:C9
Active virtual MAC address is 0005.73a0.00c9
Local virtual MAC address is 0005.73a0.00c9 (v2 IPv6 default)
Hello time 3 sec, hold time 10 sec
Next hello sent in 0.936 secs
Authentication text "ese"
Preemption enabled
Active router is FE80::212:7FFF:FEC6:8F0C, priority 120 (expires in 7.548 sec)
MAC address is 0012.7fc6.8f0c
Standby router is local
Priority 100 (default 100)
IP redundancy name is "hsrp-Fa0/0.100-201" (default)
```

Additional References

The following sections provide references related to configuring first hop redundancy protocols for IPv6.

Related Documents

Related Topic	Document Title
IPv6 link-local addresses and stateless autoconfiguration	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide

Related Topic	Document Title
IPv6 commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
Configuring HSRP in IPv4	“ Configuring HSRP ,” <i>Cisco IOS IP Application Services Configuration Guide</i>

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2281	<i>Cisco Hot Standby Router Protocol (HSRP)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **glbp ipv6**
- **ipv6 address**
- **key**
- **key chain**
- **key-string**
- **show glbp**
- **show standby**
- **standby ipv6**
- **standby preempt**
- **standby priority**
- **standby version**

Feature Information for First Hop Redundancy Protocols in IPv6

[Table 1](#) lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(15)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) document.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

[Table 1](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 **Feature Information for First Hop Redundancy Protocols for IPv6**

Feature Name	Releases	Feature Configuration Information
FHRP—GLBP Support for IPv6	12.2(14)S 12.2(33)SXI 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>GLBP protects data traffic from a failed router or circuit while allowing packet load sharing between a group of redundant routers.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • GLBP for IPv6, page 2 • Configuring and Customizing GLBP, page 7 • Customizing GLBP Configuration: Example, page 23 • GLBP Weighting: Example, page 24 • Enabling GLBP Configuration: Example, page 24
GLBP MD5 Authentication	12.2(18)S 12.3(2)T	<p>MD5 authentication provides greater security than the alternative plain text authentication scheme. MD5 authentication allows each GLBP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. A keyed hash of an incoming packet is generated and, if the hash within the incoming packet does not match the generated hash, the packet is ignored.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • Configuring GLBP Authentication, page 9 • GLBP MD5 Authentication Using Key Strings: Example, page 24 • GLBP MD5 Authentication Using Key Chains: Example, page 24 • GLBP Text Authentication: Example, page 24
IPv6 services: HSRP for IPv6	12.4(4)T 12.2(33)SRB 12.2(33)SXI	<p>The HSRP is an FHRP designed to allow for transparent failover of the first-hop IPv6 router.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • HSRP for IPv6, page 6 • Enabling an HSRP Group for IPv6 Operation, page 20

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 Secure Neighbor Discovery

First Published: February 27, 2009

Last Updated: February 27, 2009

This document provides information about configuring the Secure Neighbor Discovery (SeND) protocol for IPv6.

The SeND feature is designed to counter the threats of the Neighbor Discovery Protocol (NDP). SeND defines a set of neighbor discovery options and two neighbor discovery messages. SeND also defines a new autoconfiguration mechanism to establish address ownership.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing SeND for IPv6”](#) section on page 36.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS, image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing SeND for IPv6, page 2](#)
- [Information About Implementing SeND for IPv6, page 2](#)
- [How to Implement SeND for IPv6, page 7](#)
- [Configuration Examples for Implementing SeND for IPv6, page 28](#)
- [Additional References, page 33](#)
- [Command Reference, page 34](#)
- [Feature Information for Implementing SeND for IPv6, page 36](#)
- [Glossary, page 37](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing SeND for IPv6

SeND feature is available on crypto images because it involves using cryptographic libraries.

Information About Implementing SeND for IPv6

To configure the SeND protocol for IPv6, you should understand the following concepts:

- [IPv6 Neighbor Discovery Trust Models and Threats, page 2](#)
- [SeND Protocol, page 2](#)
- [SeND Deployment Models, page 3](#)

IPv6 Neighbor Discovery Trust Models and Threats

There are three IPv6 neighbor discovery trust models, which are described as follows:

- All authenticated nodes trust each other to behave correctly at the IP layer and not to send any neighbor discovery or router discovery (RD) messages that contain false information. This model represents a situation where the nodes are under a single administration and form a closed or semiclosed group. A corporate intranet is an example of this model.
- A router trusted by the other nodes in the network to be a legitimate router that routes packets between the local network and any connected external networks. This router is trusted to behave correctly at the IP layer and not to send any neighbor discovery or RD messages that contain false information. This model represents a public network run by an operator. The clients pay the operator, have the operator's credentials, and trust the operator to provide the IP forwarding service. The clients do not trust each other to behave correctly; any other client node must be considered able to send falsified neighbor discovery and RD messages.
- A model where the nodes do not directly trust each other at the IP layer. This model is considered suitable where a trusted network operator is not available.

Nodes on the same link use NDP to discover each other's presence and link-layer addresses, to find routers, and to maintain reachability information about the paths to active neighbors. NDP is used by both hosts and routers. The original NDP specifications used IPsec to protect NDP messages. However, not many detailed instructions for using IPsec are available. The number of manually configured security associations needed for protecting NDP can be very large, which makes that approach impractical for most purposes. These threats need to be considered and eliminated.

SeND Protocol

The SeND protocol counters NDP threats. It defines a set of new ND options, and two new ND messages (Certification Path Solicitation (CPS) and Certification Path Answer (CPA)). It also defines a new autoconfiguration mechanism to be used in conjunction with the new ND options to establish address ownership.

SeND defines the mechanisms defined in the following sections for securing the NDP:

- [Cryptographically Generated Address, page 3](#)
- [Authorization Delegation Discovery, page 3](#)

Cryptographically Generated Address

Cryptographically generated addresses (CGAs) are IPv6 addresses generated from the cryptographic hash of a public key and auxiliary parameters. This provides a method for securely associating a cryptographic public key with an IPv6 address in the SeND protocol.

The node generating a CGA address must first obtain a Rivest, Shamir, and Adelman (RSA) key pair (SeND uses an RSA public/private key pair). The node then computes the interface identifier part (which is the rightmost 64 bits) and appends the result to the prefix to form the CGA address.

CGA address generation is a one-time event. A valid CGA cannot be spoofed and the CGA parameters received associated to it is reused because the message must be signed with the private key that matches the public key used for CGA generation, which only the address owner will have.

A user cannot replay the complete SeND message (including the CGA address, CGA parameters, and CGA signature) because the signature has only a limited lifetime.

Authorization Delegation Discovery

Authorization delegation discovery is used to certify the authority of routers by using a trust anchor. A trust anchor is a third party that the host trusts and to which the router has a certification path. At a basic level, the router is certified by the trust anchor. In a more complex environment, the router is certified by a user that is certified by the trust anchor. In addition to certifying the router identity (or the right for a node to act as a router), the certification path contains information about prefixes that a router is allowed to advertise in router advertisements. Authorization delegation discovery enables a node to adopt a router as its default router.

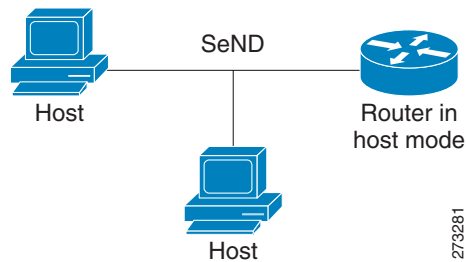
SeND Deployment Models

This section describes the following SeND deployment models:

- [Host-to-Host Deployment Without a Trust Anchor, page 3](#)
- [Neighbor Solicitation Flow, page 4](#)
- [Host-Router Deployment Model, page 5](#)
- [Router Advertisement and Certificate Path Flows, page 5](#)
- [Single CA Model, page 6](#)

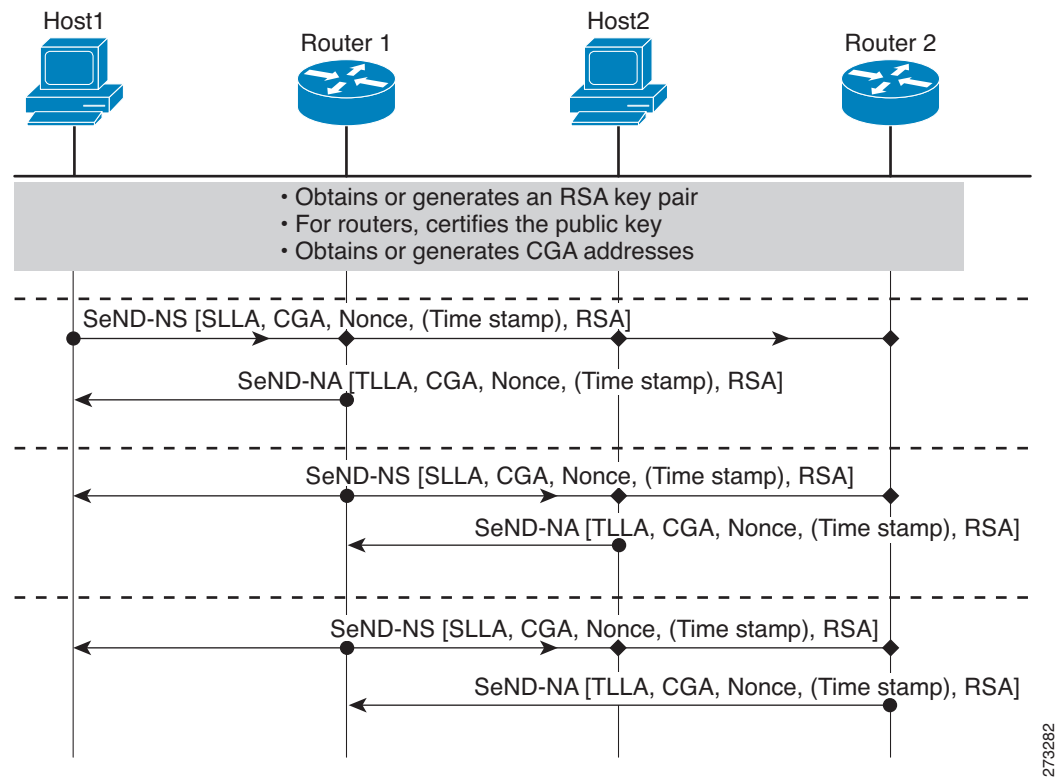
Host-to-Host Deployment Without a Trust Anchor

Deployment for SeND between hosts is straightforward. The hosts can generate a pair of RSA keys locally, autoconfigure their CGA addresses, and use them to validate their sender authority, rather than using a trust anchor to establish sender authority. [Figure 1](#) illustrates this model.

Figure 1 *Host-to-Host Deployment Model*

Neighbor Solicitation Flow

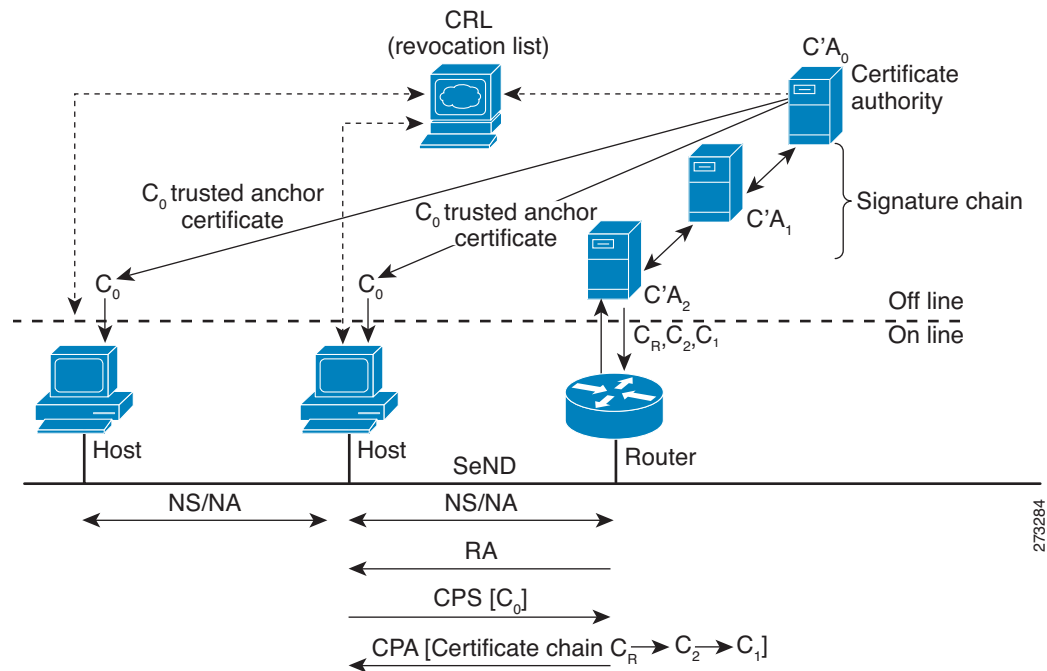
In a neighbor solicitation scenario, hosts and routers in host mode exchange neighbor solicitations and neighbor advertisements. These neighbor solicitations and neighbor advertisements are secured with CGA addresses and CGA options, and have nonce, time stamp, and RSA neighbor discovery options. Figure 2 illustrates this scenario.

Figure 2 *Neighbor Solicitation Flow*

Host-Router Deployment Model

In many cases, hosts will not have access to the infrastructure that will enable them to obtain and announce their certificates. In these situations, hosts will secure their relationship using CGA, and secure their relationship with routers using a trusted anchor. When using router advertisements (RAs), SeND mandates that routers are authenticated through a trust anchor. Figure 3 illustrates this scenario.

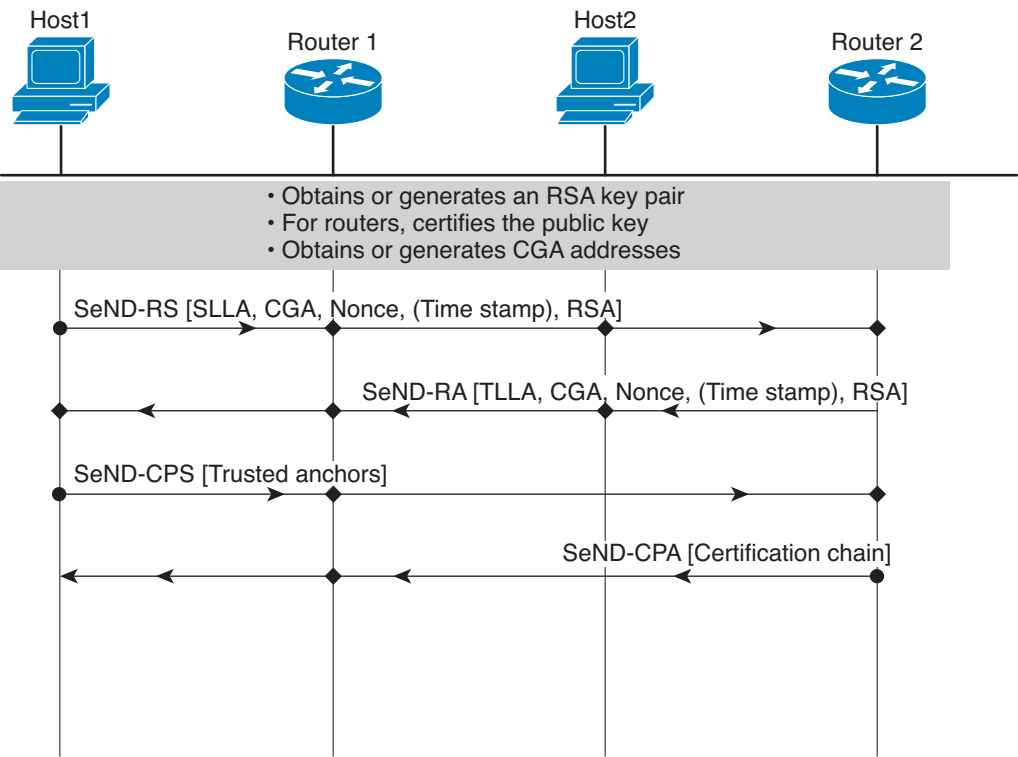
Figure 3 Host-Router Deployment Model



Router Advertisement and Certificate Path Flows

Figure 4 shows the certificate exchange performed using certification path solicitation CPS/CPA SeND messages. In the illustration, Router R is certified (using an X.509 certificate) by its own CA (certificates C_R). The CA itself (CA₂) is certified by its own CA (certificates C₂), and so on, up to a CA (CA₀) that the hosts trusts. The certificate C_R contains IP extensions per RFC 3779, which describes which prefix ranges the router R is allowed to announce (in RAs). This prefix range, certified by CA₂, is a subset of CA₂'s own range, certified by CA₁, and so on. Part of the validation process when a certification chain is received consists of validating the certification chain and the consistency of nested prefix ranges.

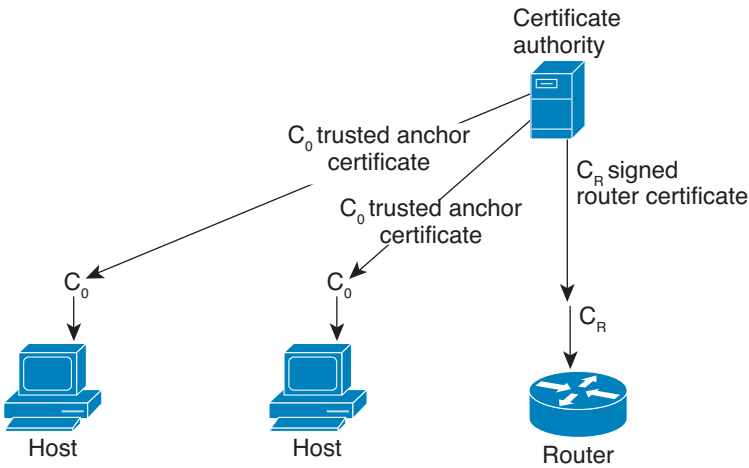
Figure 4 Router Advertisement and Certificate Path Flows



Single CA Model

The deployment model shown in Figure 3 can be simplified in an environment where both hosts and routers trust a single CA such as the Cisco certification server (CS). Figure 5 illustrates this model.

Figure 5 Single CA Deployment Model



How to Implement SeND for IPv6

The following sections describe how to configure the certificate server, the host and the router:

- [Configuring Certificate Servers to Enable SeND, page 8](#) (required)
- [Configuring a Host to Enable SeND, page 10](#) (required)
- [Configuring a Router to Enable SeND, page 12](#) (required)

To configure SeND on a Cisco device, you must understand the following concepts:

- [Certificate Servers, page 7](#)
- [Host, page 7](#)
- [Router, page 7](#)

Certificate Servers

Certificate servers are used to grant certificates after validating or certifying key pairs. A tool for granting certificates is mandatory in any SeND deployment. Many tools are available to grant certificates, for example, Open Secure Sockets Layer (OpenSSL) on Linux. However, very few certificate servers support granting certificates containing IP extensions. Cisco IOS certificate servers support every kind of certificate including the certificates containing the IP extensions. For more information on Cisco IOS certificates, refer to the [Configuring and Managing a Cisco IOS Certificate Server](#) module in the *Cisco IOS Security Configuration Guide*.

Host

SeND is available in host mode. The set of available functions on a host will be a subset of SeND functionality. CGA will be fully available and the prefix authorization delegation will be supported on the host side (sending CPS and receiving CPA).

To implement SeND, configure the host with the following parameters:

- An RSA key pair used to generate CGA addresses on the interface.
- A SeND modifier that is computed using the RSA key pair.
- A key on the SeND interface.
- CGAs on the SeND interface.
- A Public Key Infrastructure (PKI) trustpoint, with minimum content; for example, the URL of the certificate server. A trust anchor certificate must be provisioned on the host.

Router

SeND is available in router mode. You can use the **ipv6 unicast-routing** command to configure a node to a router. To implement SeND, configure routers with the same elements as that of the host. The routers will need to retrieve certificates of their own from a certificate server. The RSA key and subject name of the trustpoint are used to retrieve certificates from a certificate server. Once the certificate has been obtained and uploaded, the router generates a certificate request to the certificate server and installs the certificate.

Following is a summary of operations to be performed before SeND is configured on the host or router:

- Hosts are configured with one or more trust anchors.

- Hosts are configured with an RSA key pair or configured with the capability to locally generate it. Note that for hosts not establishing their own authority via a trust anchor, these keys are not certified by any CA.
- Routers are configured with RSA keys and corresponding certificate chains, or the capability to obtain these certificate chains that match the host trust anchor at some level of the chain.

While booting, hosts and routers must either retrieve or generate their CGAs. Typically, routers will autoconfigure their CGAs once and save them (along with the key pair used in the CGA operation) into their permanent storage. At a minimum, link-local addresses on a SeND interface should be CGAs. Additionally, global addresses can be CGAs.

Configuring Certificate Servers to Enable SeND

Hosts and routers must be configured with RSA key pairs and corresponding certificate chains before the SeND parameters are configured. Perform the following task to configure the certificate server to grant certificates. Once the certificate server is configured, other parameters for the certificate server can be configured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip http server**
4. **crypto pki trustpoint *name***
5. **ip-extension [multicast | unicast] {inherit [ipv4 | ipv6] | prefix *ipaddress* | range *min-ipaddress* *max-ipaddress*}**
6. **revocation-check {[crl] [none] [ocsp]}**
7. **exit**
8. **crypto pki server *name***
9. **grant auto**
10. **cdp-url *url-name***
11. **no shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	ip http server Example: Router(config)# ip http server	Configures the HTTP server.
Step 4	crypto pki trustpoint name Example: Router(config)# crypto pki trustpoint CA	(Optional) Declares the trustpoint that your certificate server should use and enters ca-trustpoint configuration mode. <ul style="list-style-type: none"> If you plan to use X.509 IP extensions, use this command. To automatically generate a CS trustpoint, go to Step 8.
Step 5	ip-extension [multicast unicast] {inherit [ipv4 ipv6] prefix ipaddress range min-ipaddress max-ipaddress} Example: Router(ca-trustpoint)# ip-extension prefix 2001:100::/32	(Optional) Specifies that the IP extensions are included in a certificate request either for enrollment or generation of a certificate authority (CA) for the Cisco IOS CA.
Step 6	revocation-check {[crl] [none] [ocsp]} Example: Router(ca-trustpoint)# revocation-check crl	(Optional) Sets one or more methods for revocation checking.
Step 7	exit Example: Router(ca-trustpoint)# exit	Returns to global configuration mode.
Step 8	crypto pki server name Example: Router(config)# crypto pki server CA	Configures the PKI server and places the router in server configuration mode.
Step 9	grant auto Example: Router(config-server)# grant auto	(Optional) Grants all certificate requests automatically.
Step 10	cdp-url url-name Example: Router(config-server)# cdp-url http://209.165.202.129/CA.crl	(Optional) Sets the URL name if the host is using a Certificate Revocation List (CRL).
Step 11	no shutdown Example: Router(config-server)# no shutdown	Enables the certificate server.

Configuring a Host to Enable SeND

SeND is available in host mode. Before you can configure SeND parameters in host mode, first configure the host using the following commands. Once the host has been configured, SeND parameters can be configured on it.

Summary Steps

1. `enable`
2. `configure terminal`
3. `crypto key generate rsa [general-keys | usage-keys | signature | encryption] [label key-label] [exportable] [modulus modulus-size] [storage devicename:] [on devicename:]`
4. `ipv6 cga modifier rsakeypair key-label sec-level {0 | 1}`
5. `crypto pki trustpoint name`
6. `enrollment [mode] [retry period minutes] [retry count number] url url [pem]`
7. `revocation-check {[crl] [none] [ocsp]}`
8. `exit`
9. `crypto pki authenticate name`
10. `ipv6 nd secured sec-level minimum value`
11. `interface type number`
12. `ipv6 cga rsakeypair key-label`
13. `ipv6 address ipv6-address/prefix-length link-local cga`
14. `ipv6 nd secured trustanchor trustanchor-name`
15. `ipv6 nd secured timestamp {delta value | fuzz value}`
16. `exit`
17. `ipv6 nd secured full-secure`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
	Example: <code>Host> enable</code>	
Step 2	<code>configure terminal</code>	Enters global configuration mode.
	Example: <code>Host# configure terminal</code>	

	Command or Action	Purpose
Step 3	crypto key generate rsa [general-keys usage-keys signature encryption] [label <i>key-label</i>] [exportable] [modulus <i>modulus-size</i>] [storage <i>devicename:</i>] [on <i>devicename:</i>] Example: Host(config)# crypto key generate rsa label SEND modulus 1024	Configures the RSA key.
Step 4	ipv6 cga modifier rsakeypair <i>key-label</i> sec-level {0 1} Example: Host(config)# ipv6 cga modifier rsakeypair SEND sec-level 1	Enables the RSA key to be used by SeND (generates the modifier).
Step 5	crypto pki trustpoint <i>name</i> Example: Host(config)# crypto pki trustpoint SEND	Specifies the node trustpoint and enters ca-trustpoint configuration mode.
Step 6	enrollment [mode] [retry period <i>minutes</i>] [retry count <i>number</i>] url <i>url</i> [pem] Example: Host(ca-trustpoint)# enrollment url http://209.165.200.254	Specifies the enrollment parameters of a CA.
Step 7	revocation-check {[crl] [none] [ocsp]} Example: Host(ca-trustpoint)# revocation-check none	Sets one or more methods of revocation.
Step 8	exit Example: Host(ca-trustpoint)# exit	Returns to global configuration mode.
Step 9	crypto pki authenticate <i>name</i> Example: Host(config)# crypto pki authenticate SEND	Authenticates the certification authority (by getting the certificate of the CA).
Step 10	ipv6 nd secured sec-level minimum <i>value</i> Example: Host(config)# ipv6 nd secured sec-level minimum 1	(Optional) Configures CGA. <ul style="list-style-type: none"> You can provide additional parameters such as security level and key size. In the example, the security level accepted by peers is configured.
Step 11	interface <i>type number</i> Example: Host(config)# interface fastethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 12	ipv6 cga rsakeypair <i>key-label</i> Example: Host(config-if)# ipv6 cga rsakeypair SEND	(Optional) Configures CGA on interfaces.
Step 13	ipv6 address <i>ipv6-address/prefix-length</i> link-local cga Example: Host(config-if)# ipv6 address FE80::260:3EFF:FE11:6770/23 link-local cga	Configures an IPv6 link-local address for the interface and enables IPv6 processing on the interface.
Step 14	ipv6 nd secured trustanchor <i>trustanchor-name</i> Example: Host(config-if)# ipv6 nd secured trustanchor SEND	(Optional) Configures trusted anchors to be preferred for certificate validation.
Step 15	ipv6 nd secured timestamp { delta <i>value</i> fuzz <i>value</i> } Example: Host(config-if)# ipv6 nd secured timestamp delta 300	(Optional) Configures the timing parameters.
Step 16	exit Example: Host(config-if)# exit	Returns to global configuration mode.
Step 17	ipv6 nd secured full-secure Example: Host(config)# ipv6 nd secured full-secure	(Optional) Configures general SeND parameters. <ul style="list-style-type: none"> In the example, secure mode is configured on SeND.

Configuring a Router to Enable SeND

SeND is available in the router mode. Before you can configure SeND parameters in router mode, first configure the router using the following commands. Once the router has been configured, the SeND parameters can be configured on it.

Summary Steps

- enable**
- configure terminal**
- crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
- ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}
- crypto pki trustpoint** *name*
- subject-name** [**attr** *tag*] [**eq** | **ne** | **co** | **nc**] *string*

7. **rsa**keypair *key-label*
8. **revocation-check** {[crl] [none] [ocsp]}
9. **exit**
10. **crypto pki authenticate** *name*
11. **crypto pki enroll** *name*
12. **ipv6 nd secured sec-level** [minimum *value*]
13. **interface** *type number*
14. **ipv6 cga rsa**keypair *key-label*
15. **ipv6 address** *ipv6-address/prefix-length* **link-local cga**
16. **ipv6 nd secured trustanchor** *trustanchor-name*
17. **ipv6 nd secured timestamp** {delta *value* | fuzz *value*}
18. **exit**
19. **ipv6 nd secured full-secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto key generate rsa [general-keys usage-keys signature encryption] [label <i>key-label</i>] [exportable] [modulus <i>modulus-size</i>] [storage <i>devicename</i> :] [on <i>devicename</i> :] Example: Router(config)# crypto key generate rsa label SEND modulus 1024	Configures the RSA key.
Step 4	ipv6 cga modifier rsa keypair <i>key-label</i> sec-level {0 1} Example: Router(config)# ipv6 cga modifier rsakeypair SEND sec-level 1	Enables the RSA key to be used by SeND (generates the modifier).
Step 5	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint SEND	Configures PKI for a single or multiple-tier CA, specifies the router trustpoint, and places the router in ca-trustpoint configuration mode.

	Command or Action	Purpose
Step 6	subject-name <i>[attr tag] [eq ne co nc] string</i> Example: Router(ca-trustpoint)# subject-name C=FR, ST=PACA, L=Example, O=Cisco, OU=NSSTG, CN=router	Creates a rule entry.
Step 7	rsakeypair <i>key-label</i> Example: Router(ca-trustpoint)# rsakeypair SEND	Binds the RSA key pair for SeND.
Step 8	revocation-check {[crl] [none] [ocsp]} Example: Router(ca-trustpoint)# revocation-check none	Sets one or more methods of revocation.
Step 9	exit Example: host(ca-truspoint)# exit	Returns to global configuration mode.
Step 10	crypto pki authenticate <i>name</i> Example: host(config)# crypto pki authenticate SEND	Authenticates the certification authority (by getting the certificate of the CA).
Step 11	crypto pki enroll <i>name</i> Example: Router(config)# crypto pki enroll SEND	Obtains the certificates for the router from the CA.
Step 12	ipv6 nd secured sec-level minimum <i>value</i> Example: Router(config)# ipv6 nd secured sec-level minimum 1	(Optional) Configures CGA and provides additional parameters such as security level and key size. <ul style="list-style-type: none"> In the example, the minimum security level that SeND accepts from its peers is configured.
Step 13	interface <i>type number</i> Example: Router(config)# interface fastethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 14	ipv6 cga rsakeypair <i>key-label</i> Example: Router(config-if)# ipv6 cga rsakeypair SEND	(Optional) Configures CGA on interfaces. <ul style="list-style-type: none"> In the example, CGA is generated.
Step 15	ipv6 address <i>ipv6-address/prefix-length</i> link-local cga Example: Router(config-if)# ipv6 address fe80::link-local cga	Configures an IPv6 link-local address for the interface and enables IPv6 processing on the interface.

	Command or Action	Purpose
Step 16	ipv6 nd secured trustanchor <i>trustpoint-name</i> Example: Router(config-if)# ipv6 nd secured trustanchor SEND	(Optional) Configures trusted anchors to be preferred for certificate validation.
Step 17	ipv6 nd secured timestamp { <i>delta value</i> <i>fuzz value</i> } Example: Router(config-if)# ipv6 nd secured timestamp delta 300	(Optional) Configures the timing parameters.
Step 18	exit Example: Router(config-if)# exit	Returns to global configuration mode.
Step 19	ipv6 nd secured full-secure Example: Router(config)# ipv6 nd secured full-secure	(Optional) Configures general SeND parameters, such as secure mode and authorization method. <ul style="list-style-type: none"> In the example, SeND security mode is enabled.

How to Implement SeND

The tasks in the following sections explain how to implement SeND:

- [Creating the RSA Key Pair and CGA Modifier for the Key Pair, page 15](#) (required)
- [Configuring Certificate Enrollment for a PKI, page 16](#) (required)
- [Configuring a Cryptographically Generated Address, page 19](#) (required)
- [Configuring SeND Parameters, page 21](#) (optional)

Creating the RSA Key Pair and CGA Modifier for the Key Pair

To create the RSA key pair and the CGA modifier for the key pair, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa** [*general-keys* | *usage-keys* | *signature* | *encryption*] [*label key-label*] [*exportable*] [*modulus modulus-size*] [*storage devicename:*] [*on devicename:*]
4. **ipv6 cga modifier rsakeypair** *key-label* *sec-level* {*0* | *1*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto key generate rsa [general-keys usage-keys signature encryption] [label <i>key-label</i>] [exportable] [modulus <i>modulus-size</i>] [storage <i>devicename:</i>] [on <i>devicename:</i>] Example: Router(config)# crypto key generate rsa label SeND	Generates RSA key pairs.
Step 4	ipv6 cga modifier rsakeypair <i>key-label</i> sec-level {0 1} Example: Router(config)# ipv6 cga modifier rsakeypair SeND sec-level 1	Generates the CGA modifier for a specified RSA key, which enables the key to be used by SeND.

Configuring Certificate Enrollment for a PKI

Certificate enrollment, which is the process of obtaining a certificate from a CA, occurs between the end host that requests the certificate and the CA. Each peer that participates in the PKI must enroll with a CA.

In IPv6, you can autoenroll or manually enroll the device certificate. The following task describes how to configure certificate enrollment for a PKI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **subject-name** [*x.500-name*]
5. **enrollment** [**mode**] [**retry period** *minutes*] [**retry count** *number*] **url** *url* [**pem**]
6. **serial-number** [**none**]
7. **auto-enroll** [*percent*] [**regenerate**]
8. **password** *string*
9. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
10. **fingerprint** *ca-fingerprint*

11. **ip-extension** [multicast | unicast] { inherit [ipv4 | ipv6] | prefix *ipaddress* | range *min-ipaddress* *max-ipaddress* }
12. **exit**
13. **crypto pki authenticate** *name*
14. **exit**
15. **copy** [/erase] [/verify | /noverify] *source-url* *destination-url*
16. **show crypto pki certificates**
17. **show crypto pki trustpoints** [status | label [status]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint trustpoint1	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 4	subject-name [<i>x.500-name</i>] Example: Router(ca-trustpoint)# subject-name name1	Specifies the subject name in the certificate request.
Step 5	enrollment [mode] [retry period <i>minutes</i>] [retry count <i>number</i>] url <i>url</i> [pem] Example: Router(ca-trustpoint)# enrollment url http://name1.example.com	Specifies the URL of the CA on which your router should send certificate requests.
Step 6	serial-number [none] Example: Router(ca-trustpoint)# serial-number	(Optional) Specifies the router serial number in the certificate request.
Step 7	auto-enroll [percent] [regenerate] Example: Router(ca-trustpoint)# auto-enroll	(Optional) Enables autoenrollment, allowing you to automatically request a router certificate from the CA.

	Command or Action	Purpose
Step 8	password <i>string</i> Example: Router(ca-trustpoint)# password password1	(Optional) Specifies the revocation password for the certificate.
Step 9	rsakeypair <i>key-label</i> [<i>key-size</i> [<i>encryption-key-size</i>]] Example: Router(ca-trustpoint)# rsakeypair SEND	Specifies which key pair to associate with the certificate.
Step 10	fingerprint <i>ca-fingerprint</i> Example: Router(ca-trustpoint)# fingerprint 12EF53FA 355CD23E 12EF53FA 355CD23E	(Optional) Specifies a fingerprint that can be matched against the fingerprint of a CA certificate during authentication.
Step 11	ip-extension [multicast unicast] { inherit [ipv4 ipv6] prefix <i>ipaddress</i> range <i>min-ipaddress max-ipaddress</i> } Example: Router(ca-trustpoint)# ip-extension unicast prefix 2001:100:1::/48	Add IP extensions (IPv6 prefixes or range) to verify prefix list the router is allowed to advertise.
Step 12	exit Example: Router(ca-trustpoint)# exit	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 13	crypto pki authenticate <i>name</i> Example: Router(config)# crypto pki authenticate name1	Retrieves and authenticates the CA certificate. <ul style="list-style-type: none"> This command is optional if the CA certificate is already loaded into the configuration.
Step 14	exit Example: Router(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.
Step 15	copy [/erase] [/verify /noverify] <i>source-url destination-url</i> Example: Router# copy system:running-config nvram:startup-config	(Optional) Copies the running configuration to the NVRAM startup configuration.

	Command or Action	Purpose
Step 16	<code>show crypto pki certificates</code> Example: Router# show crypto pki certificates	(Optional) Displays information about your certificates, the certificates of the CA, and RA certificates.
Step 17	<code>show crypto pki trustpoints [status label [status]]</code> Example: Router# show crypto pki trustpoints name1	(Optional) Displays the trustpoints configured in the router.

Configuring a Cryptographically Generated Address

To configure a CGA, perform the following tasks:

- [Configuring General CGA Parameters, page 19](#) (required)
- [Configuring CGA Address Generation on an Interface, page 20](#) (required)

Configuring General CGA Parameters

To configure general CGA parameters such as security level and key size, perform the following task.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ipv6 nd secured sec-level [minimum value]`
4. `ipv6 nd secured key-length [[minimum | maximum] value]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ipv6 nd secured sec-level [minimum <i>value</i>] Example: Router(config)# ipv6 nd secured sec-level minimum 1	Configures the SeND security level.
Step 4	ipv6 nd secured key-length [[minimum maximum] <i>value</i>] Example: Router(config)# ipv6 nd secured key-length minimum 512	Configures SeND key-length options.

Configuring CGA Address Generation on an Interface

To configure CGA address generation on an interface, perform the following task.

SUMMARY STEPS

- 1. **enable**
- 2. **configure terminal**
- 3. **interface** *type number*
- 4. **ipv6 cga rsa***keypair* *key-label*
- 5. **ipv6 address** {*ipv6-address/prefix-length* [**cga**] | *prefix-name sub-bits/prefix-length* [**cga**]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.

	Command or Action	Purpose
Step 4	ipv6 cga rsakeypair <i>key-label</i> Example: Router(config-if)# ipv6 cga rsakeypair SEND	Specifies which RSA key pair should be used on a specified interface.
Step 5	ipv6 address { <i>ipv6-address/prefix-length</i> [cga] <i>prefix-name sub-bits/prefix-length</i> [cga] } Example: Router(config-if)# ipv6 address 2001:0DB8:1:1::/64 cga	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface. • The cga keyword generates a CGA address. Note The CGA link-local addresses must be configured by using the ipv6 address link-local command.

Configuring SeND Parameters

To configure SeND, perform the following tasks:

- [Configuring the SeND Trustpoint, page 21](#) (optional)
- [Configuring SeND Trust Anchors on the Interface, page 24](#) (optional)
- [Configuring Secured and Nonsecured Neighbor Discovery Message Coexistence Mode, page 25](#) (optional)
- [Configuring SeND Parameters Globally, page 26](#) (optional)
- [Configuring the SeND Time Stamp, page 27](#) (optional)

Configuring the SeND Trustpoint

In the router mode, the key pair used to generate the CGA addresses on an interface must be certified by the CA and the certificate sent on demand over the SeND protocol. One RSA key pair and associated certificate is enough for SeND to operate; however, users may use several keys, identified by different labels. SeND and CGA refer to a key directly by label or indirectly by trustpoint.

Multiple steps are required to bind SeND to a trustpoint. First a key pair is generated. Then the device refers to it in a trustpoint. Then the SeND interface configuration points to the trustpoint. There are two reasons for the multiple steps:

- The same key pair can be used on several SeND interfaces
- The trustpoint contains additional information, such as the certificate, required for SeND to perform authorization delegation

A CA certificate must be uploaded for the referred trustpoint. The referred trustpoint is in reality a trusted anchor.

Several trustpoints can be configured, pointing to the same RSA keys, on a given interface. This function is useful if different hosts have different trusted anchors (that is, CAs that they trust). The router can then provide each host with the certificate signed by the CA they trust.

To configure the SeND trustpoint, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto key generate rsa** [**general-keys** | **usage-keys** | **signature** | **encryption**] [**label** *key-label*] [**exportable**] [**modulus** *modulus-size*] [**storage** *devicename:*] [**on** *devicename:*]
4. **ipv6 cga modifier rsakeypair** *key-label* **sec-level** {**0** | **1**}
5. **crypto pki trustpoint** *name*
6. **subject-name** [*x.500-name*]
7. **rsakeypair** *key-label* [*key-size* [*encryption-key-size*]]
8. **enrollment terminal** [**pem**]
9. **ip-extension** [**multicast** | **unicast**] {**inherit** [**ipv4** | **ipv6**] | **prefix** *ipaddress* | **range** *min-ipaddress* *max-ipaddress*}
10. **exit**
11. **crypto pki authenticate** *name*
12. **crypto pki enroll** *name*
13. **crypto pki import** *name* **certificate**
14. **interface** *type number*
15. **ipv6 nd secured trustpoint** *trustpoint-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto key generate rsa [general-keys usage-keys signature encryption] [label <i>key-label</i>] [exportable] [modulus <i>modulus-size</i>] [storage <i>devicename:</i>] [on <i>devicename:</i>] Example: Router(config)# crypto key generate rsa label SEND	Generates RSA key pairs.
Step 4	ipv6 cga modifier rsakeypair <i>key-label</i> sec-level { 0 1 } Example: Router(config)# ipv6 cga modifier rsakeypair SEND sec-level 1	Generates the CGA modifier for a specified RSA key, which enables the key to be used by SeND.

	Command or Action	Purpose
Step 5	crypto pki trustpoint <i>name</i> Example: Router(config)# crypto pki trustpoint trustpoint1	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 6	subject-name [<i>x.500-name</i>] Example: Router(ca-trustpoint)# subject-name name1	Specifies the subject name in the certificate request.
Step 7	rsakeypair <i>key-label</i> [<i>key-size</i> [<i>encryption-key-size</i>]] Example: Router(ca-trustpoint)# rsakeypair SEND	Specifies which key pair to associate with the certificate.
Step 8	enrollment terminal [<i>pem</i>] Example: Router(ca-trustpoint)# enrollment terminal	Specifies manual cut-and-paste certificate enrollment.
Step 9	ip-extension [<i>multicast</i> <i>unicast</i>] { <i>inherit</i> [<i>ipv4</i> <i>ipv6</i>] <i>prefix</i> <i>ipaddress</i> <i>range</i> <i>min-ipaddress</i> <i>max-ipaddress</i> } Example: Router(ca-trustpoint)# ip-extension unicast prefix 2001:100:1::/48	Adds IP extensions to the router certificate request.
Step 10	exit Example: Router(ca-trustpoint)# exit	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 11	crypto pki authenticate <i>name</i> Example: Router(config)# crypto pki authenticate trustpoint1	Authenticates the certification authority (by getting the certificate of the CA).
Step 12	crypto pki enroll <i>name</i> Example: Router(config)# crypto pki enroll trustpoint1	Obtains the certificates for your router from the CA.
Step 13	crypto pki import <i>name</i> certificate Example: Router(config)# crypto pki import trustpoint1 certificate	Imports a certificate manually via TFTP or as a cut-and-paste at the terminal.

	Command or Action	Purpose
Step 14	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 15	ipv6 nd secured trustpoint <i>trustpoint-name</i> Example: Router(config-if)# ipv6 nd secured trustpoint trustpoint1	Enables SeND on an interface and specifies which trustpoint should be used.

Configuring SeND Trust Anchors on the Interface

This task can be performed only in host mode. The host must be configured with one or more trust anchors. As soon as SeND is bound to a trustpoint on an interface (see [“Configuring the SeND Trustpoint” section on page 21](#)), this trustpoint is also a trust anchor.

A trust anchor configuration consists of the following items:

- A public key signature algorithm and associated public key, which may include parameters
- A name
- An optional public key identifier
- An optional list of address ranges for which the trust anchor is authorized

Because PKI has already been configured, the trust anchor configuration is accomplished by binding SeND to one or several PKI trustpoints. PKI is used to upload the corresponding certificates, which contain the required parameters (such as name and key).

This is an optional task. It allows you to select trust anchors listed in the CPS when requesting for a certificate. If you opt not to configure trust anchors, all the PKI trustpoints configured on the host will be considered.

To configure a trusted anchor on the interface perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto pki trustpoint** *name*
4. **enrollment terminal** [pem]
5. **exit**
6. **crypto pki authenticate** *name*
7. **interface** *type number*
8. **ipv6 nd secured trustanchor** *trustanchor-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	crypto pki trustpoint name Example: Router(config)# crypto pki trustpoint anchor1	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 4	enrollment terminal [pem] Example: Router(ca-trustpoint)# enrollment terminal	Specifies manual cut-and-paste certificate enrollment.
Step 5	exit Example: Router(ca-trustpoint)# exit	Returns to global configuration.
Step 6	crypto pki authenticate name Example: Router(config)# crypto pki authenticate anchor1	Authenticates the certification authority (by getting the certificate of the CA).
Step 7	interface type number Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	ipv6 nd secured trustanchor trustanchor-name Example: Router(config-if)# ipv6 nd secured trustanchor anchor1	Specifies a trusted anchor on an interface and binds SeND to a trustpoint.

Configuring Secured and Nonsecured Neighbor Discovery Message Coexistence Mode

During the transition to SeND secured interfaces, network operators may want to run a particular interface with a mixture of nodes accepting secured and unsecured neighbor discovery messages. To configure the coexistence mode for secure and nonsecure neighbor discovery messages on the same interface, perform the following task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd secured trustpoint** *trustpoint-name*
5. **no ipv6 nd secured full-secure**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 nd secured trustpoint <i>trustpoint-name</i> Example: Router(config-if)# ipv6 nd secured trustpoint trustpoint1	Enables SeND on an interface and specifies which trustpoint should be used.
Step 5	no ipv6 nd secured full-secure Example: Router(config-if)# no ipv6 nd secured full-secure	Provides the coexistence mode for secure and nonsecure neighbor discovery messages on the same interface.

Configuring SeND Parameters Globally

To configure SeND parameters globally, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nd secured key-length** *[[minimum | maximum] value]*
4. **ipv6 nd secured sec-level** *minimum value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 nd secured key-length [[minimum maximum] value]	Configures the SeND key-length options.
	Example: Router(config)# ipv6 nd secured key-length minimum 512	
Step 4	ipv6 nd secured sec-level minimum value	Configures the minimum security level value that can be accepted from peers.
	Example: Router(config)# ipv6 nd secured sec-level minimum 2	

Configuring the SeND Time Stamp

To configure SeND time stamp on an interface, perform the following optional task.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nd secured timestamp** {delta value | fuzz value}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 nd secured timestamp { delta <i>value</i> fuzz <i>value</i> } Example: Router(config-if)# ipv6 nd secured timestamp delta 600	Configures the SeND time stamp.

Configuration Examples for Implementing SeND for IPv6

This section provides the following configuration examples:

- [Configuring Certificate Servers: Example, page 28](#)
- [Configuring a Host to Enable SeND: Example, page 29](#)
- [Configuring a Router to Enable SeND: Example, page 30](#)
- [Configuring a SeND Trustpoint in Router Mode: Example, page 32](#)
- [Configuring SeND Trust Anchors in the Host Mode: Example, page 32](#)
- [Configuring CGA Address Generation on an Interface: Example, page 32](#)

Configuring Certificate Servers: Example

The following example shows how to configure certificate servers:

```
crypto pki server CA
 issuer-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=CA0 lifetime ca-certificate
 700 !
crypto pki trustpoint CA
 ip-extension prefix 2001::/16
 revocation-check crl
 rsakeypair CA
 no shutdown
```



Note

If you need to configure certificate servers without IP extensions, do not use the **ip-extension** command.

To display the certificate servers with IP extensions, use the **show crypto pki certificates verbose** command:

```
Router# show crypto pki certificates verbose

CA Certificate
Status: Available
Version: 3
Certificate Serial Number (hex): 01
Certificate Usage: Signature
Issuer:
  c=FR
  st=fr
```



```

l=example
o=cisco
ou=nsstg
cn=CA0
Subject:
  c=FR
  st=fr
  l=example
  o=cisco
  ou=nsstg
  cn=CA0
Validity Date:
  start date: 09:50:52 GMT Feb 5 2009
  end   date: 09:50:52 GMT Jan 6 2011
Subject Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
Signature Algorithm: MD5 with RSA Encryption
Fingerprint MD5: 87DB764F 29367A65 D05CEE3D C12E0AC3
Fingerprint SHA1: 04A06602 86AA72E9 43F2DB33 4A7D40A2 E2ED3325
X509v3 extensions:
  X509v3 Key Usage: 86000000
    Digital Signature
    Key Cert Sign
    CRL Signature
  X509v3 Subject Key ID: 75B477C6 B2CA7BBE C7866657 57C84A32 90CEFB5A
  X509v3 Basic Constraints:
    CA: TRUE
  X509v3 Authority Key ID: 75B477C6 B2CA7BBE C7866657 57C84A32 90CEFB5A
  Authority Info Access:
  X509v3 IP Extension:
    IPv6:
      2001::/16
  Associated Trustpoints: CA

```

Configuring a Host to Enable SeND: Example

The following example shows how to configure a host to enable SeND:

```

enable
configure terminal
  crypto key generate rsa label SEND modulus 1024
The name for the keys will be: SEND
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
  ipv6 cga modifier rsakeypair SEND sec-level 1
  crypto pki trustpoint SEND
  enrollment url http://209.165.200.254
  revocation-check none
  exit
  crypto pki authenticate SEND
Certificate has the following attributes:
  Fingerprint MD5: FC99580D 0A280EB4 2EB9E72B 941E9BDA
  Fingerprint SHA1: 22B10EF0 9A519177 145EA4F6 73667837 3A154C53
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
  ipv6 nd secured sec-level minimum 1
  interface fastethernet 0/0
    ipv6 cga rsakeypair SEND
    ipv6 address FE80::260:3EFF:FE11:6770 link-local cga
    ipv6 nd secured trustanchor SEND
    ipv6 nd secured timestamp delta 300

```

```
exit
ipv6 nd secured full-secure
```

To verify the configuration use the **show running-config** command:

```
host# show running-config

Building configuration...
[snip]
crypto pki trustpoint SEND
  enrollment url http://209.165.200.225
  revocation-check none
!
interface Ethernet1/0
  ip address 209.165.202.129 255.255.255.0
  duplex half
  ipv6 cga rsakeypair SEND
  ipv6 address 2001:100::/64 cga
```

Configuring a Router to Enable SeND: Example

The following example shows how to configure the router to enable SeND:

```
enable
configure terminal
crypto key generate rsa label SEND modulus 1024
ipv6 cga modifier rsakeypair SEND sec-level 1
crypto pki trustpoint SEND
  subject-name C=FR, ST=PACA, L=Example, O=Cisco, OU=NSSTG, CN=router
  rsakeypair SEND
  revocation-check none
exit
crypto pki authenticate SEND
Certificate has the following attributes:
  Fingerprint MD5: FC99580D 0A280EB4 2EB9E72B 941E9BDA
  Fingerprint SHA1: 22B10EF0 9A519177 145EA4F6 73667837 3A154C53
% Do you accept this certificate? [yes/no]: yes
Trustpoint CA certificate accepted.
crypto pki enroll SEND
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:
Re-enter password:

% The subject name in the certificate will include: C=FR, ST=fr, L=example, O=cisco,
OU=nsstg, CN=route r % The subject name in the certificate will include: Router % Include
the router serial number in the subject name? [yes/no]: no % Include an IP address in the
subject name? [no]:

Request certificate from CA? [yes/no]: yes % Certificate request sent to Certificate
Authority % The 'show crypto pki certificate SEND verbose' command will show the
fingerprint.

*Feb  5 09:40:37.171: CRYPTO_PKI: Certificate Request Fingerprint MD5:
A6892F9F 23561949 4CE96BB8 CBC85 E64
*Feb  5 09:40:37.175: CRYPTO_PKI: Certificate Request Fingerprint SHA1:
30832A66 E6EB37DF E578911D 383F 96A0 B30152E7
*Feb  5 09:40:39.843: %PKI-6-CERTRET: Certificate received from Certificate Authority
```

```
interface fastethernet 0/0
ipv6 nd secured sec-level minimum 1
ipv6 cga rsakeypair SEND
ipv6 address fe80::link-local cga
ipv6 nd secured trustanchor SEND
ipv6 nd secured timestamp delta 300
exit
ipv6 nd secured full-secure
```

To verify that the certificates are generated, use the **show crypto pki certificates** command:

Router# **show crypto pki certificates**

```
Certificate
  Status: Available
  Certificate Serial Number: 0x15
  Certificate Usage: General Purpose
  Issuer:
    cn=CA
  Subject:
    Name: Router
    hostname=Router
    c=FR
    st=fr
    l=example
    o=cisco
    ou=nsstg
    cn=router
  Validity Date:
    start date: 09:40:38 UTC Feb 5 2009
    end   date: 09:40:38 UTC Feb 5 2010
  Associated Trustpoints: SEND

CA Certificate
  Status: Available
  Certificate Serial Number: 0x1
  Certificate Usage: Signature
  Issuer:
    cn=CA
  Subject:
    cn=CA
  Validity Date:
    start date: 10:54:53 UTC Jun 20 2008
    end   date: 10:54:53 UTC Jun 20 2011
  Associated Trustpoints: SEND
```

To verify the configuration, use the **show running-config** command:

Router# **show running-config**

```
Building configuration...
[snip]
crypto pki trustpoint SEND
  enrollment url http://209.165.201.1
  subject-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=router  revocation-check none
rsakeypair SEND !
interface Ethernet1/0
  ip address 209.165.200.225 255.255.255.0
  duplex half
  ipv6 cga rsakeypair SEND
  ipv6 address FE80:: link-local cga
  ipv6 address 2001:100::/64 cga
```

Configuring a SeND Trustpoint in Router Mode: Example

The following example shows how to configure a SeND trustpoint in router mode:

```
enable
configure terminal
crypto key generate rsa label SEND
  Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.
How many bits in the modulus [512]: 778
% Generating 778 bit RSA keys, keys will be non-exportable...[OK]
ipv6 cga modifier rsakeypair SEND sec-level 1
crypto pki trustpoint trstpt1
  subject-name C=FR, ST=fr, L=example, O=cisco, OU=nsstg, CN=sa14-72b
  rsakeypair SEND
  enrollment terminal
  ip-extension unicast prefix 2001:100:1::/48
exit
crypto pki authenticate trstpt1
crypto pki enroll trstpt1
crypto pki import trstpt1 certificate
interface Ethernet 0/0
  ipv6 nd secured trustpoint trstpt1
```

Configuring SeND Trust Anchors in the Host Mode: Example

The following example shows how to configure SeND trust anchors on an interface in the host mode:

```
enable
configure terminal
! Configure the location of the CS we trust !
crypto pki trustpoint B1
  enrollment terminal
  crypto pki authenticate anchor1
exit
! Only Query a certificate signed by the CS at B2 on this interface !
interface Ethernet0/0
  ip address 204.209.1.54 255.255.255.0
  ipv6 cga rsakeypair SEND
  ipv6 address 2001:100::/64 cga
  ipv6 nd secured trustanchor anchor1
```

Configuring CGA Address Generation on an Interface: Example

The following example shows how to configure CGA address generation on an interface:

```
enable
configure terminal
interface fastEthernet 0/0
  ipv6 cga rsakeypair SEND
  ipv6 address 2001:100::/64 cga
exit
```

Additional References

The following sections provide references related to the Implementing SeND for IPv6 feature.

Related Documents

Related Topic	Document Title
Configuring certificate enrollment for a PKI	“ Configuring Certificate Enrollment for a PKI ” module in the <i>Cisco IOS Security Configuration Guide</i>

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3756	<i>IPv6 Neighbor Discovery (ND) Trust Models and Threats</i>
RFC 3779	<i>X.509 Extensions for IP Addresses and AS Identifiers</i>
RFC 3971	<i>Secure Neighbor Discovery (SeND)</i>
RFC 3972	<i>Cryptographically Generated Addresses (CGA)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **auto-enroll**
- **crypto key generate rsa**
- **crypto pki authenticate**
- **crypto pki enroll**
- **crypto pki import**
- **enrollment url (ca-trustpoint)**
- **fingerprint**
- **ip-extension**
- **ip http server**
- **ipv6 address**
- **ipv6 address link-local**
- **ipv6 cga modifier rsakeypair**
- **ipv6 cga modifier rsakeypair (interface)**
- **ipv6 nd secured certificate-db**
- **ipv6 nd secured full-secure**
- **ipv6 nd secured full-secure (interface)**
- **ipv6 nd secured key-length**
- **ipv6 nd secured sec-level**

- **ipv6 nd secured timestamp**
- **ipv6 nd secured timestamp-db**
- **ipv6 nd secured trustanchor**
- **ipv6 nd secured trustpoint**
- **password (ca-trustpoint)**
- **rsakeypair**
- **serial-number (ca-trustpoint)**
- **show ipv6 cga address-db**
- **show ipv6 cga modifier-db**
- **show ipv6 nd secured certificates**
- **show ipv6 nd secured counters interface**
- **show ipv6 nd secured nonce-db**
- **show ipv6 nd secured timestamp-db**

Feature Information for Implementing SeND for IPv6

[Table 1](#) lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(15)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) document.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

[Table 1](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 *Feature Information for Implementing IPv6 Secure Neighbor Discovery*

Feature Name	Releases	Feature Information
Secure Neighbor Discovery for Cisco IOS Software	12.4(24)T	<p>The Secure Neighbor Discovery (SeND) protocol is designed to counter the threats of the Neighbor Discovery Protocol. SeND defines a set of neighbor discovery options and two neighbor discovery messages. SeND also defines a new autoconfiguration mechanism to establish address ownership.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • How to Implement SeND for IPv6, page 7. <p>The following commands were introduced:</p> <p>ipv6 cga modifier rsakeypair, ipv6 cga modifier rsakeypair (interface), ipv6 nd secured certificate-db, ipv6 nd secured full-secure, ipv6 nd secured full-secure (interface), ipv6 nd secured key-length, ipv6 nd secured sec-level, ipv6 nd secured timestamp, ipv6 nd secured timestamp-db, ipv6 nd secured trustanchor, ipv6 nd secured trustpoint, show ipv6 cga address-db, show ipv6 cga modifier-db, show ipv6 nd secured certificates, show ipv6 nd secured counters interface, show ipv6 nd secured nonce-db, show ipv6 nd secured timestamp-db.</p> <p>The following commands were modified:</p> <p>auto-enroll, crypto key generate rsa, crypto pki authenticate, crypto pki enroll, crypto pki import, enrollment terminal (ca-trustpoint), enrollment url (ca-trustpoint), fingerprint, ip-http server, ipv6 address, ipv6 address link-local, password (ca-trustpoint), revocation-check, rsakeypair, serial-number (ca-trustpoint), subject-name.</p>

Glossary

- **CA**—certification authority.
- **CGA**—cryptographically generated address.
- **CPA**—certificate path answer.
- **CPR**—certificate path response.
- **CPS**—certification path solicitation. The solicitation message used in the addressing process.
- **CRL**—certificate revocation list.
- **CS**—certification server.
- **CSR**—certificate signing request.
- **DAD**—duplicate address detection. A mechanism that ensures two IPv6 nodes on the same link are not using the same address.

- **DER**—distinguished encoding rules. An encoding scheme for data values.
- **nonce**—An unpredictable random or pseudorandom number generated by a node and used once. In SeND, nonces are used to assure that a particular advertisement is linked to the solicitation that triggered it.
- **non-SeND node**—An IPv6 node that does not implement SeND but uses only the Neighbor Discovery protocol without security.
- **NUD**—neighbor unreachability detection. A mechanism used for tracking neighbor reachability.
- **PKI**—public key infrastructure.
- **Router Authorization Certificate**—A public key certificate.
- **RD**—Router discovery allows the hosts to discover what routers exist on the link and what subnet prefixes are available. Router discovery is a part of the Neighbor Discovery protocol.
- **SeND node**—An IPv6 node that implements SeND.
- **trust anchor**—A trust anchor is an entity that the host trusts to authorize routers to act as routers. Hosts are configured with a set of trust anchors to protect router discovery.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 over Frame Relay

First Published: April 16, 2008

Last Updated: November 10, 2008

This document describes IPv6 over Frame Relay encapsulation and multilink Frame Relay (MFR) encapsulation and provides information about QoS on Engine 3 and Engine 5 line cards on Cisco 12000 series Internet routers.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPv6 over Frame Relay”](#) section on page 17.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

-
-
-
- [How to Implement IPv6 over Frame Relay, page 6](#)
- [Configuration Examples for Implementing IPv6 over Frame Relay, page 12](#)
- [Additional References, page 14](#)
- [Command Reference, page 15](#)
- [Feature Information for Implementing IPv6 over Frame Relay, page 17](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2008–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing IPv6 over Frame Relay

-
- Before the IPv6 Provider Edge Router over MPLS (6PE) feature can be implemented, MPLS must be running over the core IPv4 network. If Cisco routers are used, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled for both IPv4 and IPv6 protocols. This module assumes that you are familiar with MPLS.
- QoS for IPv6 must be enabled. See the [Implementing QoS for IPv6](#)

[Implementing MPLS over IPv6](#)

Restrictions for Implementing IPv6 over Frame Relay

-
-
-
-

Cisco IOS Release 12.0(33)S

- Frame Relay services are supported on channelized and nonchannelized shared port adapters (SPAs).
- Engine 3:
 - Channelized OC-12—Frame Relay and MFR
 - 4-port channelized OC-12/DS3—Frame Relay
 - 4-port OC-3X Packet over SONET (PoS)—Frame Relay
 - 8-port OC-3X PoS—Frame Relay
 - 16-port OC-3X PoS—Frame Relay
 - 4-port OC-12X PoS—Frame Relay
 - 1-port OC- 48X PoS—Frame Relay
- Engine 5:
 - 8XT1/E1 SPA—Frame Relay and MFR
 - 2/4xCT-3 SPA—Frame Relay and MFR
 - 1-port channelized STM-1/OC-3 SPA—Frame Relay and MFR
 - 2/4xT3 SPA—Frame Relay
 - OC-3/OC-12 PoS SPA—Frame Relay

-
- OC-192 PoS SPA—Frame Relay

For ingress 1c/1D and nC/nD policies, primary-level hierarchical policing is not supported for IPv6 traffic.

Information About Implementing IPv6 over Frame Relay

-
-
-
-

Frame Relay Overview

Frame Relay Encapsulation on the IPv6 Interface

Frame Relay Encapsulation on Supported IPv6 Subinterfaces

-
-
-
-
-
-
-
-
-
-
-
- Graceful removal of bundle links from bundle operation
- Interfacing with layer management functions

6PE Implementation of IPv6 over Frame Relay

QoS Services Models Used with Frame Relay

QoS nC/nD Model

QoS 1c/1D Model

QoS 1C/nD Model

How to Implement IPv6 over Frame Relay

-
-
-
-

Enabling Frame Relay Switching

SUMMARY STEPS

1. enable
 configure terminal
3. frame-relay switching

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>	
	Example: <code>Router> enable</code>	
	<code>configure terminal</code>	
	<code>Router# configure terminal</code>	
	<code>frame-relay switching</code>	
	<code>Router(config)# frame-relay switching</code>	

Configuring Frame Relay on the Main IPv6 Interface

SUMMARY STEPS

- 1.
- 2.
3. *type number*

4. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits prefix-length*}

5. []
[|]

	Enables privileged EXEC mode. Enter your password if prompted.
	Enters global configuration mode.
<i>type number</i>	
Router(config)# interface pos 1/0/0	
ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits prefix-length</i> }	
Router(config-if)# ipv6 address my-prefix 2001:0DB8:0:7272::72/64	
[]	
Router(config-if)# encapsulation frame-relay	
[]	
Router(config-if)# frame-relay intf-type dce	

The following task describes how to configure Frame Relay on an IPv6 subinterface. This task can be performed on as many subinterfaces as necessary to configure your system.

[|]

8. *dlci* *cir* *virtual-template-name*

	Enables privileged EXEC mode. Enter your password if prompted.
	Enters global configuration mode.
	Specifies an interface type and number.
	Enables Frame Relay encapsulation on a specified interface.
	Configures a Frame Relay switch type.
Router(config-if)# interface pos1/0/ 0.1	Specifies a subinterface type and number, and places the router in interface configuration mode.
<pre> { } </pre>	
Router(config-if)# ipv6 address my-prefix 2001:0DB8:0:7273::72/64	
<pre> dlci cir virtual-template-name </pre>	

Creating a Virtual Interface for an MFR Bundle

Prerequisites

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8. frame-relay multilink bid

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">•
Step 2	Example:	
Step 3	Example:	

Adding Serial Interfaces to the MFR Bundle

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

Router> enable	
Router# configure terminal	
Router(config)# interface Serial1/0/0/13:0	
Router(config-if)# no ipv6 address	
Router(config-if)# no ip directed-broadcast	
Router(config-if)# encapsulation frame-relay mfr 1	
Router(config-if)# frame-relay multilink lid lid1	
<i>seconds</i>	
Router(config-if)# frame-relay multilink hello 15	
Router(config-if)# frame-relay multilink ack 6	

	Command or Action	Purpose
Step 10	Example:	
Step 11	Example:	
Step 12	Example:	

Monitoring and Maintaining MFR for IPv6

SUMMARY STEPS

1.
2.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">
Step 2	Example:	

Configuration Examples for Implementing IPv6 over Frame Relay

- -
 -

-
-

Enabling Frame Relay Switching: Example

```
Current configuration : 3910 bytes
```

```
!  
.  
.  
.  
frame-relay switching  
.  
.  
.
```

```
frame-relay switching
```

```
interface pos 1/0/0  
ipv6 address 2001:0DB8:B00:1::1/64  
encapsulation frame-relay  
frame-relay intf-type dce
```

```
frame-relay switching
```

```
interface pos 1/0/0  
no ipv6 address  
encapsulation frame-relay  
frame-relay intf-type dce
```

```
interface pos 1/0/0.1 point-to-point  
ipv6 address 2001:0DB8:A00:1::1/64  
frame-relay interface-dlci 40
```

```
interface pos 1/0/0.2 point-to-point  
ipv6 address 2001:0DB8:C058:6301::/128  
frame-relay interface-dlci 50
```

Implementing Multilink Frame Relay for IPv6: Example

```
interface MFR1
  ip address 10.25.25.1 255.255.255.252
  no ip directed-broadcast
  frame-relay interface-dlci 101
  frame-relay intf-type dce

interface Serial1/0/0/13:0
  no ip address
  no ip directed-broadcast
  encapsulation frame-relay MFR1
  no ip mroute-cache
  no arp frame-relay
```

Related Documents

Related Topic	Document Title
	<i>Cisco IOS Wide-Area Networking Configuration Guide</i>
	<i>Cisco IOS Wide-Area Networking Configuration Guide</i>
	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i>
	<i>Implementing MPLS over IPv6</i>
	<i>Implementing Multiprotocol BGP for IPv6</i>
	<i>Implementing QoS for IPv6</i>
	<i>Cisco IOS IPv6 Command Reference</i>
	<i>Cisco IOS Wide-Area Networking Command Reference</i>

Standard	Title

MIBs

MIB	MIBs Link

RFCs

RFC	Title
	<i>Multiprotocol Interconnect over Frame Relay</i>
	<i>Transmission of IPv6 Packets over Frame Relay Networks Specification</i>

Technical Assistance

Command Reference

http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about

Cisco IOS Master Command List, All Releases

Start Here:

Cisco IOS Software Release Specifics for IPv6 Features



Table 1 **Feature Information for Implementing IPv6 over Frame Relay**

Feature Name	Releases	Feature Information

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPsec in IPv6 Security

First Published: November 3, 2003

Last Updated: May 5, 2008

Cisco IOS IPv6 security features for your Cisco networking devices can protect your network against degradation or failure and also against data loss or compromise resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Cisco IOS IPsec functionality provides network data encryption at the IP packet level, offering a robust, standards-based security solution. IPsec provides data authentication and anti-replay services in addition to data confidentiality services.

IPsec is a mandatory component of IPv6 specification. OSPF for IPv6 provides IPsec authentication support and protection, and IPv6 IPsec tunnel mode and encapsulation is used to protect IPv6 unicast and multicast traffic. This document provides information about implementing IPsec in IPv6 security.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPsec in IPv6 Security” section on page 23](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IPsec for IPv6 Security, page 2](#)
- [Information About Implementing IPsec for IPv6 Security, page 2](#)
- [How to Implement IPsec for IPv6 Security, page 4](#)
- [Configuration Examples for IPsec for IPv6 Security, page 19](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2003–2009 Cisco Systems, Inc. All rights reserved.

- [Additional References, page 20](#)
- [Command Reference, page 21](#)
- [Feature Information for Implementing IPsec in IPv6 Security, page 23](#)

Prerequisites for Implementing IPsec for IPv6 Security

- You should be familiar with IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for IPv4 configuration and command reference information.
- You should be familiar with IPv6 addressing and basic configuration. Refer to [Implementing IPv6 Addressing and Basic Connectivity](#) for more information.

Information About Implementing IPsec for IPv6 Security

To implement security features for IPv6, you need to understand the following concepts:

- [OSPF for IPv6 Authentication Support with IPsec, page 2](#)
- [IPsec for IPv6, page 3](#)

OSPF for IPv6 Authentication Support with IPsec

In order to ensure that OSPF for IPv6 packets are not altered and re-sent to the router, causing the router to behave in a way not desired by its managers, OSPF for IPv6 packets must be authenticated. OSPF for IPv6 uses the IP security (IPsec) secure socket application program interface (API) to add authentication to OSPF for IPv6 packets. This API has been extended to provide support for IPv6.

OSPF for IPv6 requires the use of IPsec to enable authentication. Crypto images are required to use authentication, because only crypto images include the IPsec API needed for use with OSPF for IPv6.

In OSPF for IPv6, authentication fields have been removed from OSPF headers. When OSPF runs on IPv6, OSPF relies on the IPv6 authentication header (AH) and IPv6 encapsulating security payload (ESP) to ensure integrity, authentication, and confidentiality of routing exchanges. IPv6 AH and ESP extension headers can be used to provide authentication and confidentiality to OSPF for IPv6.

To configure IPsec, users configure a security policy, which is a combination of the security policy index (SPI) and the key (the key creates and validates the Message Digest 5 [MD5] value). IPsec for OSPF for IPv6 can be configured on an interface or on an OSPF area. For higher security, users should configure a different policy on each interface configured with IPsec. If a user configures IPsec for an OSPF area, the policy is applied to all of the interfaces in that area, except for the interfaces that have IPsec configured directly. Once IPsec is configured for OSPF for IPv6, IPsec is invisible to the user.

For information about configuring IPsec on OSPF in IPv6, see [Implementing OSPF for IPv6](#).

IPsec for IPv6

IP Security, or IPsec, is a framework of open standards developed by the Internet Engineering Task Force (IETF) that provide security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers), such as Cisco routers. IPsec provides the following optional network security services. In general, local security policy will dictate the use of one or more of these services:

- Data confidentiality—The IPsec sender can encrypt packets before sending them across a network.
- Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.
- Data origin authentication—The IPsec receiver can authenticate the source of the IPsec packets sent. This service depends upon the data integrity service.
- Antireplay—The IPsec receiver can detect and reject replayed packets.

With IPsec, data can be sent across a public network without observation, modification, or spoofing. IPsec functionality is similar in both IPv6 and IPv4; however, site-to-site tunnel mode only is supported in IPv6.

In IPv6, IPsec is implemented using the AH authentication header and the ESP extension header. The authentication header provides integrity and authentication of the source. It also provides optional protection against replayed packets. The authentication header protects the integrity of most of the IP header fields and authenticates the source through a signature-based algorithm. The ESP header provides confidentiality, authentication of the source, connectionless integrity of the inner packet, antireplay, and limited traffic flow confidentiality.

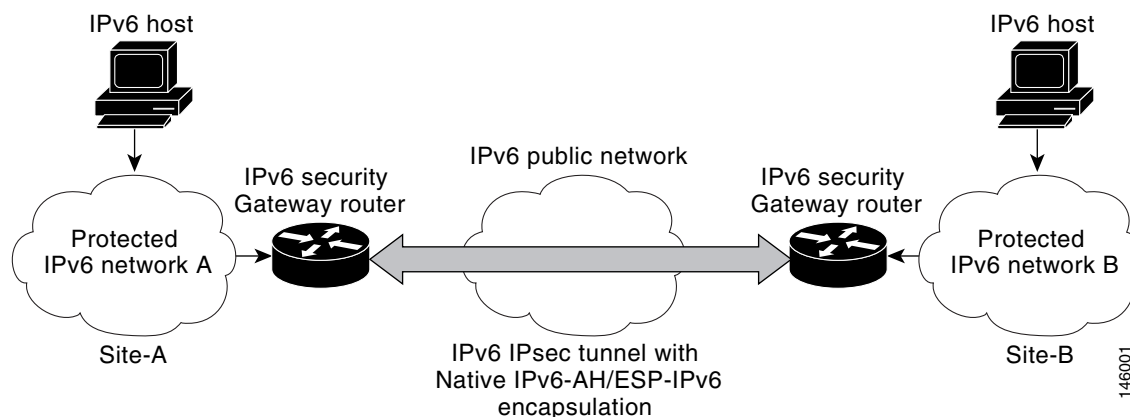
The Internet Key Exchange (IKE) protocol is a key management protocol standard that is used in conjunction with IPsec. IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

IKE is a hybrid protocol that implements the Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE) (see [Figure 1](#)). This functionality is similar to the security gateway model using IPv4 IPsec protection.

IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface

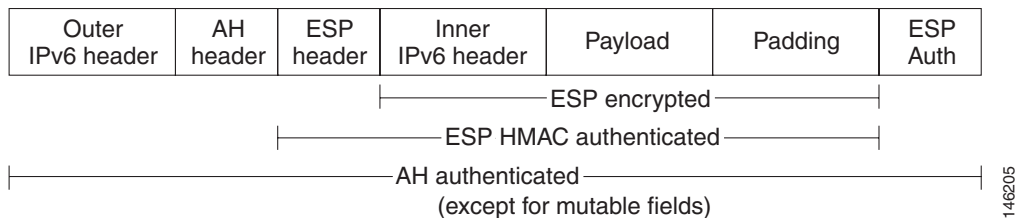
The IPsec virtual tunnel interface (VTI) provides site-to-site IPv6 crypto protection of IPv6 traffic. Native IPv6 IPsec encapsulation is used to protect all types of IPv6 unicast and multicast traffic.

The IPsec VTI allows IPv6 routers to work as security gateways, establish IPsec tunnels between other security gateway routers, and provide crypto IPsec protection for traffic from internal networks when it is sent across the public IPv6 Internet (see [Figure 1](#)). This functionality is similar to the security gateway model using IPv4 IPsec protection.

Figure 1 *IPsec Tunnel Interface for IPv6*

When the IPsec tunnel is configured, IKE and IPsec security associations (SAs) are negotiated and set up before the line protocol for the tunnel interface is changed to the UP state. The remote IKE peer is the same as the tunnel destination address; the local IKE peer will be the address picked from tunnel source interface which has the same IPv6 address scope as tunnel destination address.

Figure 2 shows the IPsec packet format.

Figure 2 *IPv6 IPsec Packet Format*

For further information on IPsec VTI, see the [IPsec Virtual Tunnel Interface](#) module in the *Cisco IOS Security Configuration Guide*.

How to Implement IPsec for IPv6 Security

The tasks in the following sections explain how to configure IPsec for IPv6:

- [Configuring a VTI for Site-to-Site IPv6 IPsec Protection, page 5](#) (required)
- [Verifying IPsec Tunnel Mode Configuration, page 13](#) (optional)
- [Troubleshooting IPsec for IPv6 Configuration and Operation, page 15](#) (optional)

Configuring a VTI for Site-to-Site IPv6 IPsec Protection

The following tasks describe how to configure an IPsec VTI for site-to-site IPsec protection of IPv6 unicast and multicast traffic. This feature allows the use of IPv6 IPsec encapsulation to protect IPv6 traffic.

- [Creating an IKE Policy and a Preshared Key in IPv6, page 5](#) (required)
- [Configuring ISAKMP Aggressive Mode, page 8](#) (optional)
- [Configuring an IPsec Transform Set and IPsec Profile, page 9](#) (required)
- [Configuring an ISAKMP Profile in IPv6, page 10](#) (optional)
- [Configuring IPv6 IPsec VTI, page 11](#) (required)

Creating an IKE Policy and a Preshared Key in IPv6

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.

After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these SAs apply to all subsequent IKE traffic during the negotiation.

You can configure multiple, prioritized policies on each peer—each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

**Note**

If you are interoperating with a device that supports only one of the values for a parameter, your choice is limited to the value supported by the other device. Aside from this limitation, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of security risks for your network and your tolerance for these risks.

IKE Peers Agreeing Upon a Matching IKE Policy

When the IKE negotiation begins, IKE searches for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the policies received from the other peer. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime that is less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used.)

If a match is found, IKE will complete negotiation, and IPsec security associations will be created. If no acceptable match is found, IKE refuses negotiation and IPsec will not be established.

**Note**

Depending on which authentication method is specified in a policy, additional configuration might be required. If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

ISAKMP Identity for Preshared Keys

You should set the ISAKMP identity for each peer that uses preshared keys in an IKE policy.

When two peers use IKE to establish IPsec SAs, each peer sends its identity to the remote peer. Each peer sends either its hostname or its IPv6 address, depending on how you have set the ISAKMP identity of the router.

By default, a peer's ISAKMP identity is the IPv6 address of the peer. If appropriate, you could change the identity to be the peer's hostname instead. As a general rule, set the identities of all peers the same way—either all peers should use their IPv6 addresses or all peers should use their hostnames. If some peers use their hostnames and some peers use their IPv6 addresses to identify themselves to each other, IKE negotiations could fail if the identity of a remote peer is not recognized and a DNS lookup is unable to resolve the identity.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **authentication** {*rsa-sig* | *rsa-encr* | **pre-share**}
5. **hash** {*sha* | **md5**}
6. **group** {**1** | **2** | **5**}
7. **encryption** {*des* | **3des** | *aes* | **aes 192** | **aes 256**}
8. **lifetime** *seconds*
9. **exit**
10. **crypto isakmp key** *password-type keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
11. **crypto keyring** *keyring-name* [**vrf** *fvr-f-name*]
12. **pre-shared-key** {**address** *address* [*mask*] | **hostname** *hostname* | **ipv6** {*ipv6-address* | *ipv6-prefix*}} **key** *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	crypto isakmp policy <i>priority</i> Example: Router(config)# crypto isakmp policy 15	Defines an IKE policy, and enters ISAKMP policy configuration mode. Policy number 1 indicates the policy with the highest priority. The smaller the <i>priority</i> argument value, the higher the priority.
Step 4	authentication {<i>rsa-sig</i> <i>rsa-encr</i> <i>pre-share</i>} Example: Router(config-isakmp-policy)# authentication pre-share	Specifies the authentication method within an IKE policy. The rsa-sig and rsa-encr keywords are not supported in IPv6.
Step 5	hash {<i>sha</i> <i>md5</i>} Example: Router(config-isakmp-policy)# hash md5	Specifies the hash algorithm within an IKE policy.
Step 6	group {1 2 5} Example: Router(config-isakmp-policy)# group 2	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 7	encryption {<i>des</i> <i>3des</i> <i>aes</i> <i>aes 192</i> <i>aes 256</i>} Example: Router(config-isakmp-policy)# encryption 3des	Specifies the encryption algorithm within an IKE policy.
Step 8	lifetime <i>seconds</i> Example: Router(config-isakmp-policy)# lifetime 43200	Specifies the lifetime of an IKE SA. Setting the IKE lifetime value is optional.
Step 9	exit Example: Router(config-isakmp-policy)# exit	Enter this command to exit ISAKMP policy configuration mode and enter global configuration mode.
Step 10	crypto isakmp key <i>enc-type-digit</i> <i>keystring</i> {<i>address</i> <i>peer-address</i> [<i>mask</i>] <i>ipv6</i> {<i>ipv6-address/ipv6-prefix</i> <i>hostname</i> <i>hostname</i>} [<i>no-xauth</i>] Example: Router(config)# crypto isakmp key 0 my-preshare-key-0 address ipv6 3ffe:1001::2/128	Configures a preshared authentication key.

	Command or Action	Purpose
Step 11	crypto keyring <i>keyring-name</i> [vrf <i>fvrfr-name</i>] Example: Router(config)# crypto keyring keyring1	Defines a crypto keyring to be used during IKE authentication.
Step 12	pre-shared-key { address <i>address</i> [<i>mask</i>] hostname <i>hostname</i> ipv6 { <i>ipv6-address</i> <i>ipv6-prefix</i> }} key <i>key</i> Example: Router (config-keyring)# pre-shared-key ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128	Defines a preshared key to be used for IKE authentication.

Configuring ISAKMP Aggressive Mode

This optional task describes how to configure ISAKMP aggressive mode.



Note

You likely do not need to configure aggressive mode in a site-to-site scenario. The default mode is typically used.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp peer** {**address** {*ipv4-address* | **ipv6** *ipv6-address* *ipv6-prefix-length*} | **hostname** *fqdn-hostname*}
4. **set aggressive-mode client-endpoint** {*client-endpoint* | **ipv6** *ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	crypto isakmp peer {address {ipv4-address ipv6 ipv6-address ipv6-prefix-length} hostname fqdn-hostname} Example: Router(config)# crypto isakmp peer address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128	Enables an IPsec peer for IKE querying for tunnel attributes.
Step 4	set aggressive-mode client-endpoint {client-endpoint ipv6 ipv6-address} Example: Router(config-isakmp-peer)# set aggressive mode client-endpoint ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128	Defines the remote peer's IPv6 address, which will be used by aggressive mode negotiation. The remote peer's address is usually the client side's end-point address.

Configuring an IPsec Transform Set and IPsec Profile

This task describes how to configure an IPsec transform set. A transform set is a combination of security protocols and algorithms that is acceptable to the IPsec routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* *transform1* [*transform2*] [*transform3*] [*transform4*]
4. **crypto ipsec profile** *name*
5. **set transform-set** *transform-set-name* [*transform-set-name2*...*transform-set-name6*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	crypto ipsec transform-set <i>transform-set-name</i> <i>transform1</i> [<i>transform2</i>] [<i>transform3</i>] [<i>transform4</i>] Example: Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des	Defines a transform set, and places the router in crypto transform configuration mode.
Step 4	crypto ipsec profile <i>name</i> Example: Router(config)# crypto ipsec profile profile0	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers.
Step 5	set transform-set <i>transform-set-name</i> [<i>transform-set-name2</i> ... <i>transform-set-name6</i>] Example: Router (config-crypto-transform)# set-transform-set myset0	Specifies which transform sets can be used with the crypto map entry.

Configuring an ISAKMP Profile in IPv6

This optional task describes how to configure an ISAKMP profile in IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp profile** *profile-name* [**accounting** *aaalist*]
4. **self-identity** {**address** | **address ipv6**} | **fqdn** | **user-fqdn** *user-fqdn*}
5. **match identity** {**group** *group-name* | **address** {*address* [*mask*] [*fvrfl*] | **ipv6** *ipv6-address*} | **host** *host-name* | **host domain** *domain-name* | **user** *user-fqdn* | **user domain** *domain-name*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	crypto isakmp profile <i>profile-name</i> [accounting <i>aaalist</i>] Example: Router(config)# crypto ipsec profile profile1	Defines an ISAKMP profile and audits IPsec user sessions.
Step 4	self-identity { address address ipv6 } fqdn user-fqdn <i>user-fqdn</i> Example: Router(config-isakmp-profile)# self-identity address ipv6	Defines the identity that the local IKE uses to identify itself to the remote peer.
Step 5	match identity { group <i>group-name</i> address { address [<i>mask</i>] [<i>fvrfl</i>] ipv6 <i>ipv6-address</i> } host <i>host-name</i> host domain <i>domain-name</i> user <i>user-fqdn</i> user domain <i>domain-name</i> } Example: Router(config-isakmp-profile)# match identity address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128	Matches an identity from a remote peer in an ISAKMP profile.

Configuring IPv6 IPsec VTI

This task describes how to configure and enable IPv6 IPsec virtual tunnel mode for IPv6.

Prerequisites

Use the **ipv6 unicast-routing** command to enable IPv6 unicast routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface tunnel** *tunnel-number*
5. **ipv6 address** *ipv6-address/prefix*
6. **ipv6 enable**
7. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
8. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
9. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** [**decapsulate-any**] | **ipsec ipv4** | **iptalk** | **ipv6** | **ipsec ipv6** | **mpls** | **nos** | **rbsep**}
10. **tunnel protection ipsec profile** *name* [**shared**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables IPv6 unicast routing. You only need to enable IPv6 unicast routing once, not matter how many interface tunnels you want to configure.
Step 4	interface tunnel <i>tunnel-number</i> Example: Router(config)# interface tunnel 0	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 5	ipv6 address <i>ipv6-address/prefix</i> Example: Router(config-if)# ipv6 address 3FFE:C000:0:7::/64 eui-64	Provides an IPv6 address to this tunnel interface, so that IPv6 traffic can be routed to this tunnel.
Step 6	ipv6 enable Example: Router(config-if)# ipv6 enable	Enables IPv6 on this tunnel interface.
Step 7	tunnel source { <i>ip-address</i> <i>ipv6-address</i> <i>interface-type interface-number</i> } Example: Router(config-if)# tunnel source ethernet0	Sets the source address for a tunnel interface.
Step 8	tunnel destination { <i>host-name</i> <i>ip-address</i> <i>ipv6-address</i> } Example: Router(config-if)# tunnel destination 2001:0DB8:1111:2222::1	Specifies the destination for a tunnel interface.

	Command or Action	Purpose
Step 9	<pre>tunnel mode {aurp cayman dvmrp eon gre gre multipoint gre ipv6 ipip [decapsulate-any] ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbscp}</pre> <p>Example: Router(config-if)# tunnel mode ipsec ipv6</p>	Sets the encapsulation mode for the tunnel interface. For IPsec, only the ipsec ipv6 keywords are supported.
Step 10	<pre>tunnel protection ipsec profile name [shared]</pre> <p>Example: Router(config-if)# tunnel protection ipsec profile profile1</p>	Associates a tunnel interface with an IPsec profile. IPv6 does not support the shared keyword.

Verifying IPsec Tunnel Mode Configuration

This optional task describes how to display information to verify IPsec tunnel mode configuration. Use the following commands as needed to verify configuration and operation.

SUMMARY STEPS

1. **show adjacency** [**summary** [*interface-type interface-number*]] | [*prefix*] [*interface interface-number*] [*connectionid id*] [**link** {**ipv4** | **ipv6** | **mpls**}] [**detail**]
2. **show crypto engine** {**accelerator** | **brief** | **configuration** | **connections** [**active** | **dh** | **dropped-packet** | **show**] | **qos**}
3. **show crypto ipsec sa** [**ipv6**] [*interface-type interface-number*] [**detailed**]
4. **show crypto isakmp peer** [**config** | **detail**]
5. **show crypto isakmp policy**
6. **show crypto isakmp profile** [**tag** *profilename* | **vrf** *vrfname*]
7. **show crypto map** [*interface interface* | **tag** *map-name*]
8. **show crypto session** [**detail**] | [**local** *ip-address* [**port** *local-port*] | [**remote** *ip-address* [**port** *remote-port*]] | **detail**] | [**fvfr** *vrf-name* | [**ivrf** *vrf-name*]]
9. **show crypto socket**
10. **show ipv6 access-list** [*access-list-name*]
11. **show ipv6 cef** [**vrf**] [*ipv6-prefix/prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]
12. **show interface** *type number* **stats**

DETAILED STEPS

	Command or Action	Purpose
Step 1	show adjacency [summary [<i>interface-type interface-number</i>]] [prefix] [interface <i>interface-number</i>] [connectionid id] [link { ipv4 ipv6 mpls }] [detail] Example: Router# show adjacency detail	Displays information about the Cisco Express Forwarding adjacency table or the hardware Layer 3-switching adjacency table.
Step 2	show crypto engine { accelerator brief configuration connections [active dh dropped-packet show] qos } Example: Router# show crypto engine connection active	Displays a summary of the configuration information for the crypto engines.
Step 3	show crypto ipsec sa [ipv6] [<i>interface-type interface-number</i>] [detailed] Example: Router# show crypto ipsec sa ipv6	Displays the settings used by current SAs in IPv6.
Step 4	show crypto isakmp peer [config detail] Example: Router# show crypto isakmp peer detail	Displays peer descriptions.
Step 5	show crypto isakmp policy Example: Router# show crypto isakmp policy	Displays the parameters for each IKE policy.
Step 6	show crypto isakmp profile [tag <i>profilename</i> vrf <i>vrfname</i>] Example: Router# show crypto isakmp profile	Lists all the ISAKMP profiles that are defined on a router.
Step 7	show crypto map [interface <i>interface</i> tag <i>map-name</i>] Example: Router# show crypto map	Displays the crypto map configuration. The crypto maps shown in this command output are dynamically generated. The user does not have to configure crypto maps.

	Command or Action	Purpose
Step 8	show crypto session [detail] [local <i>ip-address</i> [port <i>local-port</i>] [remote <i>ip-address</i> [port <i>remote-port</i>]] detail] fvfr <i>vrf-name</i> ivrf <i>vrf-name</i> Example: Router# show crypto session	Displays status information for active crypto sessions. IPv6 does not support the fvfr or ivrf keywords or the <i>vrf-name</i> argument.
Step 9	show crypto socket Example: Router# show crypto socket	Lists crypto sockets.
Step 10	show ipv6 access-list [<i>access-list-name</i>] Example: Router# show ipv6 access-list	Displays the contents of all current IPv6 access lists.
Step 11	show ipv6 cef [<i>ipv6-prefix/prefix-length</i>] [<i>interface-type interface-number</i>] [longer-prefixes similar-prefixes detail internal platform epoch source]] Example: Router# show ipv6 cef	Displays entries in the IPv6 Forwarding Information Base (FIB).
Step 12	show interface <i>type number</i> stats Example: Router# show interface fddi 3/0/0 stats	Displays numbers of packets that were process switched, fast switched, and distributed switched.


Troubleshooting IPsec for IPv6 Configuration and Operation

This optional task explains how to display information to troubleshoot the configuration and operation of IPv6 IPsec. Use the following commands only as needed to verify configuration and operation.

SUMMARY STEPS

1. **enable**
2. **debug crypto ipsec** [**error**]
3. **debug crypto engine packet** [**detail**] [**error**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router# enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	debug crypto ipsec Example: Router# debug crypto ipsec	Displays IPsec network events.
Step 3	debug crypto engine packet [detail] Example: Router# debug crypto engine packet	Displays the contents of IPv6 packets. <div style="border: 1px solid black; padding: 5px;">  Caution Using this command could flood the system and increase CPU usage if several packets are being encrypted. </div>

Examples

This section provides the following output examples:

- [Sample Output from the show crypto ipsec sa Command, page 16](#)
- [Sample Output from the show crypto isakmp peer Command, page 17](#)
- [Sample Output from the show crypto isakmp profile Command, page 17](#)
- [Sample Output from the show crypto isakmp sa Command, page 18](#)
- [Sample Output from the show crypto map Command, page 18](#)
- [Sample Output from the show crypto session Command, page 19](#)

Sample Output from the show crypto ipsec sa Command

The following is sample output from the **show crypto ipsec sa** command:

```
Router# show crypto ipsec sa

interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::A8BB:CCFF:FE01:9002

protected vrf: (none)
local ident (addr/mask/prot/port): (::/0/0/0)
remote ident (addr/mask/prot/port): (::/0/0/0)
current_peer 3FFE:2002::A8BB:CCFF:FE01:2C02 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 133, #pkts encrypt: 133, #pkts digest: 133
  #pkts decaps: 133, #pkts decrypt: 133, #pkts verify: 133
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 60, #recv errors 0

local crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:9002,
remote crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:2C02
```

```

path mtu 1514, ip mtu 1514
current outbound spi: 0x28551D9A(676666778)

inbound esp sas:
  spi: 0x2104850C(553944332)
    transform: esp-des ,
    in use settings =(Tunnel, )
    conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397507/148)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

inbound ah sas:
  spi: 0x967698CB(2524354763)
    transform: ah-sha-hmac ,
    in use settings =(Tunnel, )
    conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397507/147)
    replay detection support: Y
    Status: ACTIVE

inbound pcg sas:

outbound esp sas:
  spi: 0x28551D9A(676666778)
    transform: esp-des ,
    in use settings =(Tunnel, )
    conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397508/147)
    IV size: 8 bytes
    replay detection support: Y
    Status: ACTIVE

outbound ah sas:
  spi: 0xA83E05B5(2822636981)
    transform: ah-sha-hmac ,
    in use settings =(Tunnel, )
    conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
    sa timing: remaining key lifetime (k/sec): (4397508/147)
    replay detection support: Y
    Status: ACTIVE

outbound pcg sas:

```

Sample Output from the show crypto isakmp peer Command

The following sample output shows peer descriptions on an IPv6 router:

```
Router# show crypto isakmp peer detail
```

```

Peer: 2001:0DB8:0:1::1 Port: 500 Local: 2001:0DB8:0:2::1
Phase1 id: 2001:0DB8:0:1::1
flags:
NAS Port: 0 (Normal)
IKE SAs: 1 IPsec SA bundles: 1
last_locker: 0x141A188, last_last_locker: 0x0
last_unlocker: 0x0, last_last_unlocker: 0x0

```

Sample Output from the show crypto isakmp profile Command

The following sample output shows the ISAKMP profiles that are defined on an IPv6 router.

```
Router# show crypto isakmp profile
```

```

ISAKMP PROFILE tom
Identities matched are:
ipv6-address 2001:0DB8:0:1::1/32
Certificate maps matched are:
Identity presented is: ipv6-address fqdn
keyring(s): <none>
trustpoint(s): <all>

```

Sample Output from the show crypto isakmp sa Command

The following sample output shows the SAs of an active IPv6 device. The IPv4 device is inactive:

Router# **show crypto isakmp sa detail**

```

Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local          Remote          I-VRF      Status Encr Hash Auth DH
Lifetime Cap.

IPv6 Crypto ISAKMP SA

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1001 I-VRF:          Status: ACTIVE Encr: des Hash: sha Auth:
psk
DH: 1 Lifetime: 23:45:00 Cap: D Engine-id:Conn-id = SW:1

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1002 I-VRF:          Status: ACTIVE Encr: des Hash: sha Auth: psk
DH: 1 Lifetime: 23:45:01 Cap: D Engine-id:Conn-id = SW:2

```

Sample Output from the show crypto map Command

The following sample output shows the dynamically generated crypto maps of an active IPv6 device:

Router# **show crypto map**

```

Crypto Map "Tunnell-head-0" 65536 ipsec-isakmp
  Profile name: profile0
  Security association lifetime: 4608000 kilobytes/300 seconds
  PFS (Y/N): N
  Transform sets={
    ts,
  }

Crypto Map "Tunnell-head-0" 65537
  Map is a PROFILE INSTANCE.
  Peer = 2001:1::2

IPv6 access list Tunnell-head-0-ACL (crypto)
  permit ipv6 any any (61445999 matches) sequence 1
    Current peer: 2001:1::2
    Security association lifetime: 4608000 kilobytes/300 seconds
    PFS (Y/N): N
    Transform sets={
      ts,
    }
  Interfaces using crypto map Tunnell-head-0:
  Tunnell

```

Sample Output from the show crypto session Command

The following output from the show crypto session information provides details on currently active crypto sessions:

```
Router# show crypto session detail

Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection K - Keepalives, N -
NAT-traversal, X - IKE Extended Authentication

Interface: Tunnel1
Session status: UP-ACTIVE
Peer: 2001:1::1 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 2001:1::1
      Desc: (none)
      IKE SA: local 2001:1::2/500
              remote 2001:1::1/500 Active
              Capabilities:(none) connid:14001 lifetime:00:04:32
      IPSEC FLOW: permit ipv6 ::/0 ::/0
              Active SAs: 4, origin: crypto map
      Inbound:  #pkts dec'ed 42641 drop 0 life (KB/Sec) 4534375/72
      Outbound: #pkts enc'ed 6734980 drop 0 life (KB/Sec) 2392402/72
```

Configuration Examples for IPsec for IPv6 Security

This section provides the following configuration example:

- [Configuring a VTI for Site-to-Site IPv6 IPsec Protection: Example, page 19](#)

Configuring a VTI for Site-to-Site IPv6 IPsec Protection: Example

The following example shows configuration for a single IPv6 IPsec tunnel:

```
crypto isakmp policy 1
  authentication pre-share
!
crypto isakmp key myPreshareKey0 address ipv6 3FFE:2002::A8BB:CCFF:FE01:2C02/128
crypto isakmp keepalive 30 30
!
crypto ipsec transform-set 3des ah-sha-hmac esp-3des
!
crypto ipsec profile profile0
  set transform-set 3des
!
ipv6 cef
!
interface Tunnel0
  ipv6 address 3FFE:1001::/64 eui-64
  ipv6 enable
  ipv6 cef
  tunnel source Ethernet2/0
  tunnel destination 3FFE:2002::A8BB:CCFF:FE01:2C02
  tunnel mode ipsec ipv6
  tunnel protection ipsec profile profile0
```

Additional References

The following sections provide references related to the Implementing IPsec in IPv6 Security feature.

Related Documents

Related Topic	Document Title
OSPF for IPv6 authentication support with IPsec	<i>Implementing OSPF for IPv6</i>
IPsec VTI information	<i>IPsec Virtual Tunnel Interface</i>
IPv6 supported feature list	<i>Start Here: Cisco IOS Software Release Specifics for IPv6 Features</i>
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS IPv6 Command Reference</i>
IPv4 security configuration tasks	<i>Cisco IOS Security Configuration Guide</i>
IPv4 security commands: complete command syntax, command mode, defaults, usage guidelines, and examples	<i>Cisco IOS Security Command Reference</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2402	<i>IP Authentication Header</i>

RFCs	Title
RFC 2404	<i>The Use of Hash Message Authentication Code Federal Information Processing Standard 180-1 within Encapsulating Security Payload and Authentication Header</i>
RFC 2406	<i>IP Encapsulating Security Payload (ESP)</i>
RFC 2407	<i>The Internet Security Domain of Interpretation for ISAKMP</i>
RFC 2408	<i>Internet Security Association and Key Management Protocol (ISAKMP)</i>
RFC 2409	<i>Internet Key Exchange (IKE)</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2474	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>
RFC 3576	<i>Change of Authorization</i>
RFC 4109	<i>Algorithms for Internet Key Exchange version 1 (IKEv1)</i>
RFC 4302	<i>IP Authentication Header</i>
RFC 4306	<i>Internet Key Exchange (IKEv2) Protocol</i>
RFC 4308	<i>Cryptographic Suites for IPsec</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **authentication (IKE policy)**

- **crypto ipsec profile**
- **crypto isakmp identity**
- **crypto isakmp key**
- **crypto isakmp peer**
- **crypto isakmp policy**
- **crypto isakmp profile**
- **crypto keyring**
- **debug crypto ipv6 ipsec**
- **debug crypto ipv6 packet**
- **deny (IPv6)**
- **encryption (IKE policy)**
- **group (IKE policy)**
- **hash (IKE policy)**
- **lifetime (IKE policy)**
- **match identity**
- **permit (IPv6)**
- **pre-shared-key**
- **self-identity**
- **set aggressive-mode client-endpoint**
- **set transform-set**
- **show crypto engine**
- **show crypto ipsec policy**
- **show crypto ipsec sa**
- **show crypto isakmp key**
- **show crypto isakmp peers**
- **show crypto isakmp policy**
- **show crypto isakmp profile**
- **show crypto map (IPsec)**
- **show crypto session**
- **show crypto socket**

Feature Information for Implementing IPsec in IPv6 Security

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(4)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing IPsec in IPv6 Security

Feature Name	Releases	Feature Information
IPv6 IPsec to Authenticate Open Shortest Path First for IPv6 (OSPFv3)	12.3(4)T 12.4 12.4(2)T	OSPF for IPv6 uses the IPsec secure socket application program interface (API) to add authentication to OSPF for IPv6 packets. This API has been extended to provide support for IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> OSPF for IPv6 Authentication Support with IPsec, page 2 How to Implement IPsec for IPv6 Security, page 4
IPv6 IPsec VPN	12.4(4)T	The following sections provide information about this feature: <ul style="list-style-type: none"> Information About Implementing IPsec for IPv6 Security, page 2 How to Implement IPsec for IPv6 Security, page 4

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing IS-IS for IPv6

First Published: February 25, 2002

Last Updated: August 18, 2008

This module describes how to configure Integrated Intermediate System-to-Intermediate System (IS-IS) for IPv6. IS-IS is an Interior Gateway Protocol (IGP) that advertises link-state information throughout the network to create a picture of the network topology. IS-IS is an Open Systems Interconnection (OSI) hierarchical routing protocol that designates an intermediate system as a Level 1 or Level 2 device. Level 2 devices route between Level 1 areas to create an intradomain routing backbone. Integrated IS-IS uses a single routing algorithm to support several network address families, such as IPv6, IPv4, and OSI.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IS-IS for IPv6”](#) section on page 24.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IS-IS for IPv6, page 2](#)
- [Restrictions for Implementing IS-IS for IPv6, page 2](#)
- [Information About Implementing IS-IS for IPv6, page 2](#)
- [How to Implement IS-IS for IPv6, page 4](#)
- [Configuration Examples for IPv6 IS-IS, page 19](#)
- [Additional References, page 20](#)
- [Command Reference, page 22](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2002–2009 Cisco Systems, Inc. All rights reserved.

- [Feature Information for Implementing IS-IS for IPv6, page 24](#)

Prerequisites for Implementing IS-IS for IPv6

- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for IPv4 configuration and command reference information.
- This module assumes that you are familiar with IPv6 addressing and basic configuration. Refer to [Implementing IPv6 Addressing and Basic Connectivity](#) for more information.

Restrictions for Implementing IS-IS for IPv6

In Cisco IOS Release 12.0(22)S or later releases, and Cisco IOS Release 12.2(8)T or later releases, IS-IS support for IPv6 implements single-topology IPv6 IS-IS functionality based on IETF IS-IS WG *draft-ietf-isis-ipv6.txt*. A single shortest path first (SPF) per level is used to compute OSI, IPv4 (if configured), and IPv6 routes. The use of a single SPF means that both IPv4 IS-IS and IPv6 IS-IS routing protocols must share a common network topology. To use IS-IS for IPv4 and IPv6 routing, any interface configured for IPv4 IS-IS must also be configured for IPv6 IS-IS, and vice versa. All routers within an IS-IS area (Level 1 routing) or domain (Level 2 routing) must also support the same set of address families: IPv4 only, IPv6 only, or both IPv4 and IPv6.

Beginning with release Cisco IOS Release 12.2(15)T, IS-IS support for IPv6 is enhanced to also support multitopology IPv6 support as defined in IETF IS-IS WG *draft-ietf-isis-wg-multi-topology.txt*. Multitopology IPv6 IS-IS support uses multiple SPFs to compute routes and removes the restriction that all interfaces must support all configured address families and that all routers in an IS-IS area or domain must support the same set of address families.

The following IS-IS router configuration commands are specific to IPv4 and are not supported by, or have any effect on, IPv6 IS-IS:

- **mpls**
- **traffic-share**

Information About Implementing IS-IS for IPv6

To configure IS-IS for IPv6, you need to understand the following concept:

- [IS-IS Enhancements for IPv6, page 2](#)

IS-IS Enhancements for IPv6

IS-IS in IPv6 functions the same and offers many of the same benefits as IS-IS in IPv4. IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and OSI routes. Extensions to the IS-IS command-line interface (CLI) allow configuration of IPv6-specific parameters. IPv6 IS-IS extends the address families supported by IS-IS to include IPv6, in addition to OSI and IPv4. IS-IS in IPv6 supports either single-topology mode or multiple topology mode.

IS-IS Single-Topology Support for IPv6

Single-topology support for IPv6 allows IS-IS for IPv6 to be configured on interfaces along with other network protocols (for example, IPv4 and Connectionless Network Service [CLNS]). All interfaces must be configured with the identical set of network address families. In addition, all routers in the IS-IS area (for Level 1 routing) or the domain (for Level 2 routing) must support the identical set of network layer address families on all interfaces.

When single-topology support for IPv6 is being used, either old- or new-style TLVs may be used. However, the TLVs used to advertise reachability to IPv6 prefixes use extended metrics. Cisco routers do not allow an interface metric to be set to a value greater than 63 if the configuration is not set to support only new-style TLVs for IPv4. In single-topology IPv6 mode, the configured metric is always the same for both IPv4 and IPv6.

IS-IS Multitopology Support for IPv6

IS-IS multitopology support for IPv6 allows IS-IS to maintain a set of independent topologies within a single area or domain. This mode removes the restriction that all interfaces on which IS-IS is configured must support the identical set of network address families. It also removes the restriction that all routers in the IS-IS area (for Level 1 routing) or domain (for Level 2 routing) must support the identical set of network layer address families. Because multiple SPF calculations are performed, one for each configured topology, it is sufficient that connectivity exists among a subset of the routers in the area or domain for a given network address family to be routable.

You can use the **isis ipv6 metric** command to configure different metrics on an interface for IPv6 and IPv4.

When multitopology support for IPv6 is used, use the **metric-style wide** command to configure IS-IS to use new-style TLVs because TLVs used to advertise IPv6 information in link-state packets (LSPs) are defined to use only extended metrics.

Transition from Single-Topology to Multitopology Support for IPv6

All routers in the area or domain must use the same type of IPv6 support, either single-topology or multitopology. A router operating in multitopology mode will not recognize the ability of the single-topology mode router to support IPv6 traffic, which will lead to holes in the IPv6 topology. To transition from single-topology support to the more flexible multitopology support, a multitopology transition mode is provided.

The multitopology transition mode allows a network operating in single-topology IS-IS IPv6 support mode to continue to work while upgrading routers to include multitopology IS-IS IPv6 support. While in transition mode, both types of TLVs (single-topology and multitopology) are sent in LSPs for all configured IPv6 addresses, but the router continues to operate in single-topology mode (that is, the topological restrictions of the single-topology mode are still in effect). After all routers in the area or domain have been upgraded to support multitopology IPv6 and are operating in transition mode, transition mode can be removed from the configuration. Once all routers in the area or domain are operating in multitopology IPv6 mode, the topological restrictions of single-topology mode are no longer in effect.

IPv6 IS-IS Local RIB

A router that is running IS-IS IPv6 maintains a local RIB in which it stores all routes to destinations it has learned from its neighbors. At the end of each SPF calculation, IS-IS attempts to install the best (that is, the least-cost) routes to a destination present in the local RIB in the global IPv6 routing table.

For further information on the IPv6 IS-IS local RIB, see the [“Verifying IPv6 IS-IS Configuration and Operation”](#) section.

How to Implement IS-IS for IPv6

When configuring supported routing protocols in IPv6, you must create the routing process, enable the routing process on interfaces, and customize the routing protocol for your particular network.



Note

The following sections describe the configuration tasks for creating an IPv6 IS-IS routing process and enabling the routing process on interfaces. The following sections do not provide in-depth information on customizing IS-IS because the protocol functions the same in IPv6 as it does in IPv4. Refer to the publications referenced in the [“Related Documents”](#) section for further IPv6 and IPv4 configuration and command reference information.

The tasks in the following sections explain how to configure IPv6 IS-IS. Each task in the list is identified as either required or optional:

- [Configuring Single-Topology IS-IS for IPv6, page 4](#) (required)
- [Configuring Multitopology IS-IS for IPv6, page 6](#) (optional)
- [Customizing IPv6 IS-IS, page 7](#) (optional)
- [Redistributing Routes into an IPv6 IS-IS Routing Process, page 10](#) (optional)
- [Redistributing IPv6 IS-IS Routes Between IS-IS Levels, page 11](#) (optional)
- [Disabling IPv6 Protocol-Support Consistency Checks, page 12](#) (optional)
- [Verifying IPv6 IS-IS Configuration and Operation, page 14](#) (optional)

Configuring Single-Topology IS-IS for IPv6

This task explains how to create an IPv6 IS-IS process and enable IPv6 IS-IS support on an interface.

Configuring IS-IS comprises two activities. The first activity creates an IS-IS routing process and is performed using protocol-independent IS-IS commands. The second activity in configuring IPv6 IS-IS configures the operation of the IS-IS protocol on an interface.

Prerequisites

Before configuring the router to run IPv6 IS-IS, globally enable IPv6 using the **ipv6 unicast-routing** global configuration command. For details on basic IPv6 connectivity tasks, refer to [Implementing IPv6 Addressing and Basic Connectivity](#).

Restrictions

If you are using IS-IS single-topology support for IPv6, IPv4, or both IPv6 and IPv4, you may configure both IPv6 and IPv4 on an IS-IS interface for Level 1, Level 2, or both Level 1 and Level 2. However, if both IPv6 and IPv4 are configured on the same interface, they must be running the same IS-IS level. That is, IPv4 cannot be configured to run on IS-IS Level 1 only on a specified Ethernet interface while IPv6 is configured to run IS-IS Level 2 only on the same Ethernet interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **net** *network-entity-title*
5. **exit**
6. **interface** *type number*
7. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}
8. **ipv6 router isis** *area-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis <i>area-tag</i> Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	net <i>network-entity-title</i> Example: Router(config-router)# net 49.0001.0000.0000.000c.00	Configures an IS-IS network entity title (NET) for the routing process. <ul style="list-style-type: none"> The <i>network-entity-title</i> argument defines the area addresses for the IS-IS area and the system ID of the router. <p>Note For more details about the format of the <i>network-entity-title</i> argument, refer to the “Configuring ISO CLNS” chapter in the <i>Cisco IOS ISO CLNS Configuration Guide</i>.</p>
Step 5	exit Example: Router(config-router)# exit	Exits router configuration mode and enters global configuration mode.
Step 6	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0/1	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 7	ipv6 address {ipv6-address/prefix-length prefix-name sub-bits/prefix-length}	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
	Example: Router(config-if)# ipv6 address 2001:0DB8::3/64	Note Refer to Implementing IPv6 Addressing and Basic Connectivity for more information on configuring IPv6 addresses.
Step 8	ipv6 router isis area-name	Enables the specified IPv6 IS-IS routing process on an interface.
	Example: Router(config-if)# ipv6 router isis area2	

Configuring Multitopology IS-IS for IPv6

This task explains how to configure multitopology IS-IS in IPv6.

When multitopology IS-IS for IPv6 is configured, the **transition** keyword allows a user who is working with the single-topology SPF mode of IS-IS IPv6 to continue to work while upgrading to multitopology IS-IS. After every router is configured with the **transition** keyword, users can remove the **transition** keyword on each router. When transition mode is not enabled, IPv6 connectivity between routers operating in single-topology mode and routers operating in multitopology mode is not possible.

You can continue to use the existing IPv6 topology while upgrading to multitopology IS-IS. The optional **isis ipv6 metric** command allows you to differentiate between link costs for IPv6 and IPv4 traffic when operating in multitopology mode.

Prerequisites

Perform the following steps after you have configured IS-IS for IPv6.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** area-tag
4. **metric-style wide** [transition] [level-1 | level-2 | level-1-2]
5. **address-family ipv6** [unicast | multicast]
6. **multi-topology** [transition]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis area-tag Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	metric-style wide [transition] [level-1 level-2 level-1-2] Example: Router(config-router)# metric-style wide level-1	Configures a router running IS-IS to generate and accept only new-style TLVs.
Step 5	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 6	multi-topology [transition] Example: Router(config-router-af)# multi-topology	Enables multitopology IS-IS for IPv6. <ul style="list-style-type: none"> The optional transition keyword allows an IS-IS IPv6 user to continue to use single-topology mode while upgrading to multitopology mode.

Customizing IPv6 IS-IS

This task explains how to configure a new administrative distance for IPv6 IS-IS, configure the maximum number of equal-cost paths that IPv6 IS-IS will support, configure summary prefixes for IPv6 IS-IS, and configure an IS-IS instance to advertise the default IPv6 route (::/0). It also explains how to configure the hold-down period between partial route calculations (PRCs) and how often Cisco IOS software performs the SPF calculation when using multitopology IS-IS.

You can customize IS-IS multitopology for IPv6 for your network, but you likely will not need to do so. The defaults for this feature are set to meet the requirements of most customers and features. If you change the defaults, refer to the IPv4 configuration guide and the IPv6 command reference to find the appropriate syntax.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **default-information originate** [**route-map** *map-name*]
6. **distance** *value*
7. **maximum-paths** *number-paths*
8. **summary-prefix** *ipv6-prefix/prefix-length* [**level-1** | **level-1-2** | **level-2**]
9. **prc-interval** *seconds* [*initial-wait*] [*secondary-wait*]
10. **spf-interval** [**level-1** | **level-2**] *seconds* [*initial-wait*] [*secondary-wait*]
11. **exit**
12. **interface** *type number*
13. **isis ipv6 metric** *metric-value* [**level-1** | **level-2** | **level-1-2**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis <i>area-tag</i> Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none">The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.

	Command or Action	Purpose
Step 5	default-information originate [route-map <i>map-name</i>] Example: Router(config-router-af)# default-information originate	(Optional) Injects a default IPv6 route into an IS-IS routing domain. <ul style="list-style-type: none"> The route-map keyword and <i>map-name</i> argument specify the conditions under which the IPv6 default route is advertised. If the route map keyword is omitted, then the IPv6 default route will be unconditionally advertised at Level 2.
Step 6	distance <i>value</i> Example: Router(config-router-af)# distance 90	(Optional) Defines an administrative distance for IPv6 IS-IS routes in the IPv6 routing table. <ul style="list-style-type: none"> The <i>value</i> argument is an integer from 10 to 254. (The values 0 to 9 are reserved for internal use).
Step 7	maximum-paths <i>number-paths</i> Example: Router(config-router-af)# maximum-paths 3	(Optional) Defines the maximum number of equal-cost routes that IPv6 IS-IS can support. <ul style="list-style-type: none"> This command also supports IPv6 Border Gateway Protocol (BGP) and Routing Information Protocol (RIP). The <i>number-paths</i> argument is an integer from 1 to 64. The default for BGP is one path; the default for IS-IS and RIP is 16 paths.
Step 8	summary-prefix <i>ipv6-prefix/prefix-length</i> [level-1 level-1-2 level-2] Example: Router(config-router-af)# summary-prefix 2001:0DB8::/24	(Optional) Allows a Level 1-2 router to summarize Level 1 prefixes at Level 2, instead of advertising the Level 1 prefixes directly when the router advertises the summary. <ul style="list-style-type: none"> The <i>ipv6-prefix</i> argument in the summary-prefix command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.
Step 9	prc-interval <i>seconds</i> [<i>initial-wait</i>] [<i>secondary-wait</i>] Example: Router(config-router-af)# prc-interval 20	(Optional) Configures the hold-down period between PRCs for multipotology IS-IS for IPv6.
Step 10	spf-interval [level-1 level-2] <i>seconds</i> [<i>initial-wait</i>] [<i>secondary-wait</i>] Example: Router(config-router-af)# spf-interval 30	(Optional) Configures how often Cisco IOS software performs the SPF calculation for multipotology IS-IS for IPv6.

	Command or Action	Purpose
Step 11	exit Example: Router(config-router-af)# exit	Exits address family configuration mode, and returns the router to router configuration mode. <ul style="list-style-type: none">Repeat this step to exit router configuration mode and return the router to global configuration mode.
Step 12	interface <i>type number</i> Example: Router(config-router)# interface Ethernet 0/0/1	Specifies the interface type and number, and enters interface configuration mode.
Step 13	isis ipv6 metric <i>metric-value</i> [level-1 level-2 level-1-2] Example: Router(config-if)# isis ipv6 metric 20	(Optional) Configures the value of an multipotology IS-IS for IPv6 metric.

Redistributing Routes into an IPv6 IS-IS Routing Process

This task explains how to redistribute IPv6 routes between protocols.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **redistribute** *source-protocol* [*process-id*] [**include-connected**] [*target-protocol-options*] [*source-protocol-options*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis <i>area-tag</i> Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.

	Command or Action	Purpose
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	redistribute source-protocol [process-id] [include-connected] [target-protocol-options] [source-protocol-options] Example: Router(config-router-af)# redistribute bgp 64500 metric 100 route-map isismap	Redistributes routes from the specified protocol into the IS-IS process. <ul style="list-style-type: none"> The <i>source-protocol</i> argument can be one of the following keywords: bgp, connected, isis, rip, or static. Only the arguments and keywords relevant to this task are specified here.

Redistributing IPv6 IS-IS Routes Between IS-IS Levels

This task explains how to redistribute IPv6 routes learned at one IS-IS level into a different level.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis area-tag**
4. **address-family ipv6 [unicast | multicast]**
5. **redistribute isis [process-id] {level-1 | level-2} into {level-1 | level-2} distribute-list list-name**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis area-tag Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.

	Command or Action	Purpose
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	redistribute isis [<i>process-id</i>] { level-1 level-2 } into { level-1 level-2 } distribute-list <i>list-name</i> Example: Router(config-router-af)# redistribute isis level-1 into level-2	Redistributes IPv6 routes from one IS-IS level into another IS-IS level. <ul style="list-style-type: none"> By default, the routes learned by Level 1 instances are redistributed by the Level 2 instance. Note The <i>protocol</i> argument must be isis in this configuration of the redistribute command. Only the arguments and keywords relevant to this task are specified here.

Disabling IPv6 Protocol-Support Consistency Checks

This task explains how to disable protocol-support consistency checks in IPv6 single-topology mode.

For single-topology IS-IS IPv6, routers must be configured to run the same set of address families. IS-IS performs consistency checks on hello packets and will reject hello packets that do not have the same set of configured address families. For example, a router running IS-IS for both IPv4 and IPv6 will not form an adjacency with a router running IS-IS for IPv4 or IPv6 only. In order to allow adjacency to be formed in mismatched address-families network, the **adjacency-check** command in IPv6 address family configuration mode must be disabled. This command is designed for use only in special situations. Please read the following note before configuring this task.



Note

Entering the **no adjacency-check** command can adversely affect your network configuration. Enter the **no adjacency-check** command only when you are running IPv4 IS-IS on all your routers and you want to add IPv6 IS-IS to your network but you need to maintain all your adjacencies during the transition. When the IPv6 IS-IS configuration is complete, remove the **no adjacency-check** command from the configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis** *area-tag*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **no adjacency-check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis area-tag Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the unicast IPv6 unicast address family. By default, the router is placed in configuration mode for the unicast IPv6 address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 5	no adjacency-check Example: Router(config-router-af)# no adjacency-check	Disables the IPv6 protocol-support consistency checks performed on hello packets, allowing IPv6 to be introduced into an IPv4-only network without disrupting existing adjacencies. <ul style="list-style-type: none"> The adjacency-check command is enabled by default.

Disabling IPv4 Subnet Consistency Checks

This task explains how to disable IPv4 subnet consistency checking when forming adjacencies. Cisco IOS software historically makes checks on hello packets to ensure that the IPv4 address is present and has a consistent subnet with the neighbor from which the hello packets are received. To disable this check, use the **no adjacency-check** command in the router configuration mode. However, if multitopology IS-IS is configured, this check is automatically suppressed, because multitopology IS-IS requires routers to form an adjacency regardless of whether or not all routers on a LAN support a common protocol.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router isis area-tag**
4. **no adjacency-check**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router isis area-tag Example: Router(config)# router isis area2	Enables IS-IS for the specified IS-IS routing process, and enters router configuration mode.
Step 4	no adjacency-check Example: Router(config-router-af)# no adjacency-check	Disables the IPv6 protocol-support consistency checks performed on hello packets, allowing IPv6 to be introduced into an IPv4-only network without disrupting existing adjacencies. <ul style="list-style-type: none"> The adjacency-check command is enabled by default.

Verifying IPv6 IS-IS Configuration and Operation

This task explains how to display information to verify the configuration and operation of IPv6 IS-IS.

SUMMARY STEPS

1. **enable**
2. **show ipv6 protocols [summary]**
3. **show isis [process-tag] [ipv6 | *] topology**
4. **show clns [process-tag] neighbors [interface-type interface-number] [area] [detail]**
5. **show clns area-tag is-neighbors [type number] [detail]**
6. **show isis [process-tag] database [level-1] [level-2] [l1] [l2] [detail] [lsid]**
7. **show isis ipv6 rib [ipv6-prefix]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ipv6 protocols [summary] Example: Router# show ipv6 protocols	Displays the parameters and current state of the active IPv6 routing processes.
Step 3	show isis [process-tag] [ipv6 *] topology Example: Router# show isis topology	Displays a list of all connected routers running IS-IS in all areas.
Step 4	show clns [process-tag] neighbors [interface-type interface-number] [area] [detail] Example: Router# show clns neighbors detail	Displays end system (ES), intermediate system (IS), and multitopology IS-IS (M-ISIS) neighbors.
Step 5	show clns area-tag is-neighbors [type number] [detail] Example: Router# show clns is-neighbors detail	Displays IS-IS adjacency information for IS-IS neighbors. <ul style="list-style-type: none"> Use the detail keyword to display the IPv6 link-local addresses of the neighbors.
Step 6	show isis [process-tag] database [level-1] [level-2] [l1] [l2] [detail] [lspid] Example: Router# show isis database detail	Displays the IS-IS link-state database. <ul style="list-style-type: none"> In this example, the contents of each LSP are displayed using the detail keyword.
Step 7	show isis ipv6 rib [ipv6-prefix] Example: Router# show isis ipv6 rib	Displays the IPv6 local RIB.

Troubleshooting Tips

You can use several system debugging commands to check your IS-IS for IPv6 implementation.

If adjacencies are not coming up properly, use the **debug isis adj-packets** command.

If you are using IS-IS multitopology for IPv6 and want to display statistical information about building routes between intermediate systems, use the **debug isis spf-statistics** command.

To display a log of significant events during an IS-IS SPF computation, use the **debug isis spf-events** command.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 protocols Command](#)
- [Sample Output from the show isis topology Command](#)
- [Sample Output from the show clns is-neighbors Command](#)
- [Sample Output from the show clns neighbors Command, page 17](#)
- [Sample Output from the show isis database Command](#)
- [Sample Output from the show isis ipv6 rib Command](#)

Sample Output from the show ipv6 protocols Command

In the following example, output information about the parameters and current state of that active IPv6 routing processes is displayed using the **show ipv6 protocols** command:

```
Router# show ipv6 protocols

IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "isis"
  Interfaces:
    Ethernet0/0/3
    Ethernet0/0/1
    Serial1/0/1
    Loopback1 (Passive)
    Loopback2 (Passive)
    Loopback3 (Passive)
    Loopback4 (Passive)
    Loopback5 (Passive)
  Redistribution:
    Redistributing protocol static at level 1
  Address Summarization:
    L2: 2001:0DB8:33::/16  advertised with metric 0
    L2: 2001:0DB8:44::/16  advertised with metric 20
    L2: 2001:0DB8:66::/16  advertised with metric 10
    L2: 2001:0DB8:77::/16  advertised with metric 10
```

Sample Output from the show isis topology Command

In the following example, output information about all connected routers running IS-IS in all areas is displayed using the **show isis topology** command:

```
Router# show isis topology

IS-IS paths to level-1 routers
System Id      Metric  Next-Hop      Interface      SNPA
0000.0000.000C
0000.0000.000D  20      0000.0000.00AA Se1/0/1        *HDLC*
0000.0000.000F  10      0000.0000.000F Et0/0/1        0050.e2e5.d01d
0000.0000.00AA  10      0000.0000.00AA Se1/0/1        *HDLC*

IS-IS paths to level-2 routers
System Id      Metric  Next-Hop      Interface      SNPA
0000.0000.000A  10      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000B  20      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000C  --      --             --             --
0000.0000.000D  30      0000.0000.000A Et0/0/3        0010.f68d.f063
0000.0000.000E  30      0000.0000.000A Et0/0/3        0010.f68d.f063
```

Sample Output from the show clns is-neighbors Command

In the following example, output information to confirm that the local router has formed all the necessary IS-IS adjacencies with other IS-IS neighbors is displayed using the **show clns is-neighbors** command. To display the IPv6 link-local addresses of the neighbors, specify the **detail** keyword.

Router# **show clns is-neighbors detail**

```

System Id      Interface  State  Type Priority  Circuit Id      Format
0000.0000.00AA Sel/0/1    Up     L1   0          00             Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::YYYY:D37C:C854:5
  Uptime: 17:21:38
0000.0000.000F Et0/0/1    Up     L1   64         0000.0000.000C.02 Phase V
  Area Address(es): 49.0001
  IPv6 Address(es): FE80::XXXX:E2FF:FEE5:D01D
  Uptime: 17:21:41
0000.0000.000A Et0/0/3    Up     L2   64         0000.0000.000C.01 Phase V
  Area Address(es): 49.000b
  IPv6 Address(es): FE80::ZZZZ:F6FF:FE8D:F063
  Uptime: 17:22:06

```

Sample Output from the show clns neighbors Command

In the following example, detailed output information that displays both end system (ES) and intermediate system (IS) neighbors is displayed using the **show clns neighbors** command with the **detail** keyword.

Router# **show clns neighbors detail**

```

System Id      Interface  SNPA          State  Holdtime  Type Protocol
0000.0000.0007 Et3/3      aa00.0400.6408 UP     26        L1   IS-IS
  Area Address(es): 20
  IP Address(es): 172.16.0.42*
  Uptime: 00:21:49
0000.0C00.0C35 Et3/2      0000.0c00.0c36 Up     91        L1   IS-IS
  Area Address(es): 20
  IP Address(es): 192.168.0.42*
  Uptime: 00:21:52
0800.2B16.24EA Et3/3      aa00.0400.2d05 Up     27        L1   M-ISIS
  Area Address(es): 20
  IP Address(es): 192.168.0.42*
  IPv6 Address(es): FE80::2B0:8EFF:FE31:EC57
  Uptime: 00:00:27
0800.2B14.060E Et3/2      aa00.0400.9205 Up     8         L1   IS-IS
  Area Address(es): 20
  IP Address(es): 192.168.0.30*
  Uptime: 00:21:52

```

Sample Output from the show isis database Command

In the following example, detailed output information about LSPs received from other routers and the IPv6 prefixes they are advertising is displayed using the **show isis database** command with the **detail** keyword specified:

Router# **show isis database detail**

```

IS-IS Level-1 Link State Database
LSPID          LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C00.0C35.00-00 0x0000000C  0x5696        325           0/0/0
  Area Address: 47.0004.004D.0001
  Area Address: 39.0001
  Metric: 10   IS 0000.0C00.62E6.03
  Metric: 0    ES 0000.0C00.0C35

```

```
--More--
0000.0C00.40AF.00-00* 0x00000009 0x8452 608 1/0/0
Area Address: 47.0004.004D.0001
Topology: IPv4 (0x0) IPv6 (0x2)
NLPID: 0xCC 0x8E
IP Address: 172.16.21.49
Metric: 10 IS 0800.2B16.24EA.01
Metric: 10 IS 0000.0C00.62E6.03
Metric: 0 ES 0000.0C00.40AF
IPv6 Address: 2001:0DB8::/32
Metric: 10 IPv6 (MT-IPv6) 2001:0DB8::/64
Metric: 5 IS-Extended cisco.03
Metric: 10 IS-Extended cisco1.03
Metric: 10 IS (MT-IPv6) cisco.03

IS-IS Level-2 Link State Database:
LSPID LSP Seq Num LSP Checksum LSP Holdtime ATT/P/OL
0000.0000.000A.00-00 0x00000059 0x378A 949 0/0/0
Area Address: 49.000b
NLPID: 0x8E
IPv6 Address: 2001:0DB8:1:1:1:1:1:1
Metric: 10 IPv6 2001:0DB8:2:YYYY::/64
Metric: 10 IPv6 2001:0DB8:3:YYYY::/64
Metric: 10 IPv6 2001:0DB8:2:YYYY::/64
Metric: 10 IS-Extended 0000.0000.000A.01
Metric: 10 IS-Extended 0000.0000.000B.00
Metric: 10 IS-Extended 0000.0000.000C.01
Metric: 0 IPv6 11:1:YYYY:1:1:1:1:1/128
Metric: 0 IPv6 11:2:YYYY:1:1:1:1:1/128
Metric: 0 IPv6 11:3:YYYY:1:1:1:1:1/128
Metric: 0 IPv6 11:4:YYYY:1:1:1:1:1/128
Metric: 0 IPv6 11:5:YYYY:1:1:1:1:1/128
0000.0000.000A.01-00 0x00000050 0xB0AF 491 0/0/0
Metric: 0 IS-Extended 0000.0000.000A.00
Metric: 0 IS-Extended 0000.0000.000B.00
```

Sample Output from the show isis ipv6 rib Command

The following example shows output from the **show isis ipv6 rib** command. An asterisk (*) indicates prefixes that have been installed in the master IPv6 RIB as IS-IS routes. Following each prefix is a list of all paths in order of preference, with optimal paths listed first and suboptimal paths listed after optimal paths.

```
Router# show isis ipv6 rib
```

```
IS-IS IPv6 process "", local RIB
2001:0DB8:88:1::/64
  via FE80::210:7BFF:FEC2:ACC9/Ethernet2/0, type L2 metric 20 LSP [3/7]
  via FE80::210:7BFF:FEC2:ACCC/Ethernet2/1, type L2 metric 20 LSP [3/7]
* 2001:0DB8:1357:1::/64
  via FE80::202:7DFF:FE1A:9471/Ethernet2/1, type L2 metric 10 LSP [4/9]
* 2001:0DB8:45A::/64
  via FE80::210:7BFF:FEC2:ACC9/Ethernet2/0, type L1 metric 20 LSP [C/6]
  via FE80::210:7BFF:FEC2:ACCC/Ethernet2/1, type L1 metric 20 LSP [C/6]
  via FE80::210:7BFF:FEC2:ACC9/Ethernet2/0, type L2 metric 20 LSP [3/7]
  via FE80::210:7BFF:FEC2:ACCC/Ethernet2/1, type L2 metric 20 LSP [3/7]
```

Configuration Examples for IPv6 IS-IS

This section provides the following configuration examples:

- [Configuring Single-Topology IS-IS for IPv6: Example, page 19](#)
- [Customizing IPv6 IS-IS: Example, page 19](#)
- [Redistributing Routes into an IPv6 IS-IS Routing Process: Example, page 19](#)
- [Redistributing IPv6 IS-IS Routes Between IS-IS Levels: Example, page 20](#)
- [Disabling IPv6 Protocol-Support Consistency Checks: Example, page 20](#)
- [Configuring Multitopology IS-IS for IPv6: Example, page 20](#)
- [Configuring the IS-IS IPv6 Metric for Multitopology IS-IS: Example, page 20](#)

Configuring Single-Topology IS-IS for IPv6: Example

The following example enables single-topology mode, creates an IS-IS process, defines the NET, configures an IPv6 address on an interface, and configures the interface to run IPv6 IS-IS:

```
ipv6 unicast-routing
!
router isis
 net 49.0001.0000.0000.000c.00
 exit
interface Ethernet0/0/1
 ipv6 address 2001:0DB8::3/64
 ipv6 router isis area2
```

Customizing IPv6 IS-IS: Example

The following example advertises the IPv6 default route (::/0)—with an origin of Ethernet interface 0/0/1—with all other routes in router updates sent on Ethernet interface 0/0/1. This example also sets an administrative distance for IPv6 IS-IS to 90, defines the maximum number of equal-cost paths that IPv6 IS-IS will support as 3, and configures a summary prefix of 2001:0DB8::/24 for IPv6 IS-IS.

```
router isis
 address-family ipv6
 default-information originate
 distance 90
 maximum-paths 3
 summary-prefix 2001:0DB8::/24
 exit
```

Redistributing Routes into an IPv6 IS-IS Routing Process: Example

The following example redistributes IPv6 BGP routes into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute bgp 64500 metric 100 route-map isismap
 exit
```

Redistributing IPv6 IS-IS Routes Between IS-IS Levels: Example

The following example redistributes IPv6 IS-IS Level 1 routes into the IPv6 IS-IS Level 2 routing process:

```
router isis
 address-family ipv6
 redistribute isis level-1 into level-2
```

Disabling IPv6 Protocol-Support Consistency Checks: Example

The following example disables the **adjacency-check** command to allow a network administrator to configure IPv6 IS-IS on the router without disrupting the existing adjacencies:

```
router isis
 address-family ipv6
 no adjacency-check
```

Configuring Multitopology IS-IS for IPv6: Example

The following example configures multitopology IS-IS in IPv6 after you have configured IS-IS for IPv6:

```
router isis
 metric-style wide
 address-family ipv6
 multi-topology
```

Configuring the IS-IS IPv6 Metric for Multitopology IS-IS: Example

The following example sets the value of an IS-IS IPv6 metric to 20:

```
interface Ethernet 0/0/1
 isis ipv6 metric 20
```

Where to Go Next

If you want to implement more IPv6 routing protocols, refer to the [Implementing RIP for IPv6](#) or [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing IS-IS for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IS-IS configuration tasks	“Integrated IS-IS Feature Roadmap,” <i>Cisco IOS IP Routing Protocols Configuration Guide</i>
IS-IS commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Routing Protocols Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> CISCO-IETF-IP-FORWARD-MIB CISCO-IETF-IP-MIB 	<p>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
RFC 1195	<i>Use of OSI IS-IS for Routing in TCP/IP and Dual Environments</i>
RFC 5120	<i>M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)</i>
RFC 5308	<i>Routing IPv6 with IS-IS</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **address-family ipv6 (IS-IS)**
- **adjacency-check**
- **debug isis spf-events**
- **default-information originate (IPv6 IS-IS)**
- **distance (IPv6)**
- **ipv6 router isis**
- **isis ipv6 metric**
- **multi-topology**
- **pre-interval (IPv6)**
- **redistribute isis (IPv6)**
- **router-id (IPv6)**
- **show clns neighbors**
- **show ipv6 protocols**
- **show isis database**
- **show isis ipv6 rib**
- **show isis spf-log**
- **show isis topology**
- **spf-interval (IPv6)**

- **summary-prefix (IPv6 IS-IS)**

Feature Information for Implementing IS-IS for IPv6

Table 7 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(8)T or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 7 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 7 Feature Information for Implementing IS-IS for IPv6

Feature Name	Releases	Feature Information
IPv6 routing: route redistribution	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IS-IS for IPv6 supports redistributing routes into an IPv6 IS-IS routing process and redistributing IPv6 IS-IS routes between IS-IS levels. The following sections provide information about this feature: <ul style="list-style-type: none"> Information About Implementing IS-IS for IPv6, page 2 Redistributing Routes into an IPv6 IS-IS Routing Process, page 10 Redistributing IPv6 IS-IS Routes Between IS-IS Levels, page 11
IPv6 routing: IS-IS support for IPv6	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(8)T 12.3 12.3(2)T 12.4 12.4(2)T	IPv6 enhancements to IS-IS allow IS-IS to advertise IPv6 prefixes in addition to IPv4 and OSI routes. The following sections provide information about this feature: <ul style="list-style-type: none"> IS-IS Enhancements for IPv6, page 2 Configuring Single-Topology IS-IS for IPv6, page 4 Customizing IPv6 IS-IS, page 7 Redistributing Routes into an IPv6 IS-IS Routing Process, page 10 Redistributing IPv6 IS-IS Routes Between IS-IS Levels, page 11

Table 7 **Feature Information for Implementing IS-IS for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 routing: IS-IS multipotology support for IPv6	12.0(26)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	IS-IS multipotology support for IPv6 allows IS-IS to maintain a set of independent topologies within a single area or domain. The following sections provide information about this feature: <ul style="list-style-type: none"> • IS-IS Enhancements for IPv6, page 2 • IS-IS Multipotology Support for IPv6, page 3 • Transition from Single-Topology to Multipotology Support for IPv6, page 3 • Configuring Multipotology IS-IS for IPv6, page 6
IPv6 routing: IS-IS local RIB	12,2(22)S 12.2(33)SRA 12.2(33)SXH	A router that is running IS-IS IPv6 maintains a local RIB in which it stores all routes to destinations it has learned from its neighbors. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 IS-IS Local RIB, page 3 • Verifying IPv6 IS-IS Configuration and Operation, page 14

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 for Network Management

First Published: June 7, 2001
Last Updated: March 18, 2009

This document describes the concepts and commands used to manage Cisco applications over IPv6 and to implement IPv6 for network management. The **copy**, **ping**, **telnet**, and **traceroute** commands have been modified to provide IPv6 management capability. Secure Shell (SSH) has been enhanced to provide support for IPv6 addresses that enable a Cisco router to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

Cisco IOS IPv6 embedded management components have IPv6-compliant operability in IPv6 and hybrid IPv6 and IPv4 networks. Cisco IOS embedded management components include system message logging (syslog), Cisco Networking Services (CNS) agents, Config logger, Hypertext Transfer Protocol server (HTTP(S)), tool command language (TCL), Network Configuration Protocol (NETCONF), Service-Oriented Access Protocol (SOAP), and IP Service Level Agreements (SLAs).

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the “[Feature Information for Implementing IPv6 for Network Management](#)” section on [page 21](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

-
-
- [, page 7](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

, page 15

[Additional References](#), page 18

[Command Reference](#), page 20

[Feature Information for Implementing IPv6 for Network Management](#), page 21

Prerequisites for Implementing IPv6 for Network Management

- By default, IPv6 routing is disabled in the Cisco IOS software. To enable IPv6 routing, you must first enable the forwarding of IPv6 traffic globally on the router and then you must assign IPv6 addresses to individual interfaces in the router. At least one interface must have IPv6 configured.
- To enable Telnet access to a router, you must create a vty interface and password.
- This document assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Additional References](#)” section for IPv4 configuration and command reference information.

Information About Implementing IPv6 for Network Management

-
- [TFTP File Downloading for IPv6](#), page 2
- [ping and traceroute Commands in IPv6](#), page 3
- [SSH over an IPv6 Transport](#), page 3
- [SNMP over an IPv6 Transport](#), page 3
- [Cisco IOS IPv6 Embedded Management Components](#), page 4

Telnet Access over IPv6

TFTP File Downloading for IPv6

copy

copy

```
Router# copy running-config tftp://[3ffe:xxxx:c18:1:290:27ff:fe3a:9e9a]/running-config
```


**Note**

be enclosed in square brackets ([]) when the address is used in TFTP source or destination URLs; a literal IPv6 address specified without a port number need not be enclosed in square brackets. Refer to RFC 2732, *Format for Literal IPv6 Addresses in URLs*,

ping and traceroute Commands in IPv6

SSH over an IPv6 Transport

SSH in IPv6 functions the same and offers the same benefits as SSH in IPv4—the SSH Server feature enables an SSH client to make a secure, encrypted connection to a Cisco router and the SSH Client feature enables a Cisco router to make a secure, encrypted connection to another Cisco router or to any other device running an SSH server. IPv6 enhancements to SSH consist of support for IPv6 addresses that enable a Cisco router to accept and establish secure, encrypted connections with remote IPv6 nodes over an IPv6 transport.

SNMP over an IPv6 Transport

Cisco IOS IPv6 MIBs

Cisco has long supported IP-MIB and IP-FORWARD-MIB in IPv4. CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB are IPv6 MIBs that are defined as being protocol-independent, but are implemented only for IPv6 objects and tables. In Cisco IOS Release 12.2(33)SRC, IP-MIB and IP-FORWARD-MIB were updated to RFC 4293 and RFC 4292 standards, as follows:

The upgrade is backward-compatible; all IP-MIB and IP-FORWARD-MIB objects and tables still appear.

IP-MIB and IP-FORWARD-MIB include new IPv6-only, IPv4-only, and protocol-version independent (PVI) objects and tables. However, IPv6 supports IPv6-only and the new IPv6 part of the PVI objects and tables in these MIBs.

CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB have been removed from Cisco IOS Release 12.2(33)SRC. Information in these MIBs is now included in these new MIBs: IP-MIB and IP-FORWARD-MIB.

MIBs Supported for IPv6

-
-
-
-
-
-
-
-
-
-
-

•

-
-
-

<<Note: I guessed the exact release--please let me know if these are not correct!>>

In Cisco IOS Release 12.2(33)SXI, 12.2(33)SRD, 12.4, 12.4(24)T, 12.2(44)SE and 12.2(50)SG, an SNMP query for cdpCacheAddress provides an IPv6 global unicast address. Previously, the query returned an IPv6 link-local address.

Cisco IOS IPv6 Embedded Management Components

-
-
-
-
-
-
-
-

Syslog

IPv4-based logging host (syslog server) by providing the host's IP address in IPv4 format (for example, 192.168.0.0) or IPv6 format (for example, 2001:0DB8:A00:1::1/64).

As of Cisco IOS Release 12.4(4)T and 12.2(33)SRC, this feature is backward-compatible with existing IPv4 and new IPv6 addresses and hostnames.

CNS Agents

-
-
-
-

CNS Configuration Agent

synchronized configuration updates, and the configuration engine reports the status of the configuration load as an event to which a network monitoring or workflow application can subscribe.

CNS Event Agent

CNS EXEC Agent

“

Config Logger

-

HTTP(S) IPv6 Support

TCL

NETCONF

SOAP Message Format

Configuration Guide.

IP SLAs for IPv6

-
-
-
-
-

How to Implement IPv6 for Network Management

-
-
-
-

Enabling Telnet Access to an IPv6 Router and Establishing a Telnet Session

SUMMARY STEPS

- 1. enable
- 2. configure terminal
- 3. ipv6 host port ipv6-address1 ipv6-address2...ipv6-address4
- 4. aux | |] line-number ending-line-number
- 5. password
- 6. login local tacacs
- 7. ipv6 access-class { | }
- host [port] [keyword

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	
	Example: Router> enable	
	terminal	
	Router# configure terminal	
	ipv6 host name [port] ipv6-address1 ipv6-address2...ipv6-address4	
	Router(config)# ipv6 host cisco-sj 2001:0DB8:20:1::12	
	[] line-number ending-line-number	
	Router(config)# line vty 0 4	
	password password	
	Router(config)# password hostword	

	(Optional) Enables password checking at login.
<pre> ipv6-access-list-name { }</pre>	
<pre>Router(config)# ipv6 access-list hostlist host [port] [keyword</pre>	

Prerequisites

- -
- Implementing IPv6 Addressing and Basic Connectivity*



Restrictions

authentication mechanism supported by SSH over an IPv6 transport; the TACACS+ and RADIUS user authentication mechanisms are not supported over an IPv6 transport.



```
1 2 -c 3des aes128-cbc aes192-cbc aes256-cbc -l userid | -l userid:  
-l :rotary -m hmac-md5 | hmac-md5-96 | hmac-sha1 |  
hmac-sha1-96 -o numberofpasswordprompts -p
```

By default, five vty lines are defined (0–4); therefore, five terminal sessions are possible. After the SSH executes a shell, the vty timeout starts. The vty timeout defaults to 10 minutes.

You can also specify the number of authentication retries, not to exceed five authentication retries. The default is three.


```
Router(config)# exit
```

```
[ { | } ] [ { |  
| aes256-cbc -1 -1  
:  
:rotary -m hmac-md5  
hmac-md5-96 hmac-sha1 hmac-sha1-96 -o  
numberofpasswordprompts -p
```

Use an SNMP community string to define the relationship between the SNMP manager and the agent. The community string acts like a password to regulate access to the agent on the router. Optionally, you can specify one or more of the following characteristics associated with the string:

An access list of IP addresses of the SNMP managers that are permitted to use the community string to gain access to the agent.

A MIB view, which defines the subset of all MIB objects accessible to the given community.

Read and write or read-only permission for the MIB objects accessible to the community.

You can configure one or more community strings. To remove a specific community string, use the **no snmp-server community** command.

The **snmp-server host** command specifies which hosts will receive SNMP notifications, and whether you want the notifications sent as traps or inform requests. The **snmp-server enable traps** command globally enables the production mechanism for the specified notification types (such as Border Gateway Protocol [BGP] traps, config traps, entity traps, and Hot Standby Router Protocol [HSRP] traps).

enable

configure terminal

snmp-server community [view] [ro | rw] [ipv6] [view]
snmp-server engineID remote udp-port vrf

snmp-server group v1 v2c v3 auth noauth priv context context-name
read read-view write write-view notify notify-view access ipv6
named-access-list acl-number acl-name

snmp-server host hostname ip-address vrf vrf-name traps informs version 1 2c | 3
auth | noauth | priv community-string udp-port port notification-type



	Command	Purpose
Step 1	<div>Example:</div>	<div><ul style="list-style-type: none"></div>
Step 2	<div>Example:</div>	
Step 3	<div>Example: AAAA:BBBB:CCCC:DDDD::FFFF</div>	

Disabling HTTP Access to an IPv6 Router

SUMMARY STEPS

- 1.
- 2.
- 3.



DETAILED STEPS

enable	
Router> enable	
configure terminal	
Router# configure terminal	
no ip http server	
Router(config)# no ip http server	

Configuration Examples for Implementing IPv6 for Network Management

-
-
-

Enabling Telnet Access to an IPv6 Router Configuration: Examples

```
        ipv6 host cisco-sj 2001:0db8:20:1::12
        end
show host
```

```
Name/address lookup uses static mappings

Codes:UN - unknown, EX - expired, OK - OK, ?? - revalidate
       temp - temporary, perm - permanent
       NA - Not Applicable None - Not defined

Host          Port  Flags      Age Type  Address(es)
cisco-sj      None  (perm, OK) 0  IPv6  2001:0db8:20:1::12

Router(config)# line vty 0 4
```

```
password lab
login
```

```
Router# telnet cisco-sj
```

```
Trying cisco-sj (2001:0db8:20:1::12)... Open
```

```
User Access Verification
```

```
Password:
cisco-sj
.
.
.
verification
```

```
Router#
```

```
Router#
```

```
Router#
```

Line	User	Host(s)	Idle	Location
* 0 con 0		idle	00:00:00	
130 vty 0		idle	00:00:22	8800::3

```
Router#
```

Line	User	Host(s)	Idle	Location
* 0 con 0		idle	00:00:00	
130 vty 0		idle	00:02:47	cisco-sj

If the user at the connecting router suspends the session with ^6x and then enters the command, the IPv6 connection is displayed:

```
Router#
```

Conn	Host	Address	Byte	Idle	Conn Name
* 1	cisco-sj	2001:0db8:20:1::12	0	0	cisco-sj

The Conn Name field shows the hostname of the destination only if it is known. If it is not known, the output might look similar to the following:

```
Router#
```

Conn	Host	Address	Byte	Idle	Conn Name
* 1	2001:0db8:20:1::12	2001:0db8:20:1::12	0	0	2001:0db8:20:1::12

In the following example, the command is used to show that HTTP access is disabled on the router:

command is used to show that HTTP access is

```
Router#  
  
Building configuration...  
!  
Current configuration : 1490 bytes  
!  
version 12.2  
!  
hostname Router  
!  
no ip http server  
!  
line con 0  
line aux 0  
line vty 0 4
```

```
Router(config)#  
Router(config)#  
Router(config)# snmp-server host 172.16.1.27 version 2c public  
Router(config)# snmp-server host 172.16.1.111 version 1 public  
Router(config)# snmp-server host 3ffe:b00:c18:1::3/127 public
```

Associate an SNMP Server Group with Specified Views Example

```
snmp-server context A  
snmp mib community-map commA context A target-list commAVpn  
snmp mib target list commAVpn vrf CustomerA  
snmp-server view viewA ciscoPingMIB included  
snmp-server view viewA ipForward included  
snmp-server group GROUP1 v2c context A read viewA write viewA notify  
access ipv6 public2
```

Create an SNMP Notification Server Example

```
enable  
configure terminal  
snmp-server community mgr view restricted rw ipv6 mgr2  
snmp-server engineID remote 3ffe:b00:c18:1::3/127 remotev6  
snmp-server group public v2c access ipv6 public2  
snmp-server host host1.com 2c vrf trap-vrf  
snmp-server user user1 bldg1 remote 3ffe:b00:c18:1::3/127 v2c access ipv6  
public2  
snmp-server enable traps bgp  
exit
```

Where to Go Next

Additional References

Related Documents

Related Topic	Document Title
	<ul style="list-style-type: none"><i>IP SLAs—Analyzing IP Service Levels Using the ICMP Echo Operation</i><i>IP SLAs—Analyzing IP Service Levels Using the TCP Connect Operation</i><i>IP SLAs—Analyzing IP Service Levels Using the UDP Echo Operation</i><i>IP SLAs—Analyzing IP Service Levels Using the UDP Jitter Operation</i><i>IP SLAs—Analyzing VoIP Service Levels Using the UDP Jitter Operation</i>

Standards

Standard	Title

RFCs

[illegible]

RFC	Title
	<i>User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)</i>
	<i>Default Address Selection for Internet Protocol version 6 (IPv6)</i>
	<i>IP Forwarding Table MIB</i>
	<i>Management Information Base for the Internet Protocol (IP)</i>

Technical Assistance

[illegible]

Cisco IOS IPv6 Command Reference

http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

show ip sockets
show ipv6 routers
show ipv6 traffic
snmp-server community
snmp-server engineID remote
snmp-server group
snmp-server host
snmp-server user
ssh
telnet
traceroute

Feature Information for Implementing IPv6 for Network Management

[Start Here:](#)

[Cisco IOS Software Release Specifics for IPv6 Features](#)



Table 1 *Feature Information for Managing Cisco IOS Applications over IPv6*

Feature Name	Releases	Feature Information
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">

Feature Information for Managing Cisco IOS Applications over IPv6 (continued)

		<ul style="list-style-type: none">
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">
		<ul style="list-style-type: none">

Table 1 *Feature Information for Managing Cisco IOS Applications over IPv6 (continued)*

[illegible]

IPv6 support for TCL	12.2(33)SRC 12.4(20)T Cisco IOS XE Release 2.1	IPv6 supports TCL. The following section provides information about this feature: TCL, page 6
IPv6 NETCONF support	12.2(33)SB 12.2(33)SRC 12.4(20)T	The Network Configuration Protocol (NETCONF) defines a simple mechanism through which a network device can be managed, configuration data information can be retrieved, and new configuration data can be uploaded and manipulated. The following section provides information about this feature: NETCONF, page 6
IPv6 support in SOAP	12.2(33)SB 12.2(33)SRC 12.4(20)T Cisco IOS XE Release 2.1	SOAP is a protocol intended for exchanging structured information in a decentralized, distributed environment. The following section provides information about this feature: SOAP Message Format, page 7
IP SLAs for IPv6	12.2(33)SB 12.2(33)SRC 12.4(20)T	IP SLAs are supported for IPv6. The following section provides information about this feature: IP SLAs for IPv6, page 7

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing Mobile IPv6

First Published: March 28, 2005
Last Updated: November 11, 2008

Mobile IP is part of both IPv4 and IPv6 standards. Mobile IP allows a host device to be identified by a single IP address even though the device may move its physical point of attachment from one network to another. Regardless of movement between different networks, connectivity at the different points is achieved seamlessly without user intervention. Roaming from a wired network to a wireless or wide-area network is also done with ease. Mobile IP provides ubiquitous connectivity for users, whether they are within their enterprise networks or away from home.

This document provides information about Mobile IPv6.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Mobile IPv6” section on page 30](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Restrictions for Implementing Mobile IPv6, page 2](#)
- [Information About Implementing Mobile IPv6, page 2](#)
- [How to Implement Mobile IPv6, page 7](#)
- [Configuration Examples for Implementing Mobile IPv6, page 24](#)
- [Additional References, page 27](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2005–2008 Cisco Systems, Inc. All rights reserved.

- [Command Reference, page 28](#)
- [Feature Information for Implementing Mobile IPv6, page 30](#)

Restrictions for Implementing Mobile IPv6

When using the network mobility (NEMO) basic support protocol feature, users should not enable any IPv6 routing protocols on any of the roaming interfaces.

Information About Implementing Mobile IPv6

Before you configure Mobile IPv6 for Cisco IOS software, you should understand the following concepts:

- [Mobile IPv6 Overview, page 2](#)
- [How Mobile IPv6 Works, page 3](#)
- [IPv6 NEMO, page 3](#)
- [Mobile IPv6 Home Agent, page 3](#)
- [Packet Headers in Mobile IPv6, page 5](#)
- [IPv6 Neighbor Discovery with Mobile IPv6, page 5](#)
- [Mobile IPv6 Tunnel Optimization, page 6](#)
- [IPv6 Host Group Configuration, page 6](#)

Mobile IPv6 Overview

Mobile IPv4 provides an IPv4 node with the ability to retain the same IPv4 address and maintain uninterrupted network and application connectivity while traveling across networks. In Mobile IPv6, the IPv6 address space enables Mobile IP deployment in any kind of large environment. No foreign agent is needed to use Mobile IPv6.

System infrastructures do not need an upgrade to accept Mobile IPv6 nodes. IPv6 autoconfiguration simplifies mobile node (MN) Care of Address (CoA) assignment.

Mobile IPv6 benefits from the IPv6 protocol itself; for example, Mobile IPv6 uses IPv6 option headers (routing, destination, and mobility) and benefits from the use of neighbor discovery.

Mobile IPv6 provides optimized routing, which helps avoid triangular routing. Mobile IPv6 nodes work transparently even with nodes that do not support mobility (although these nodes do not have route optimization).

Mobile IPv6 is fully backward-compatible with existing IPv6 specifications. Therefore, any existing host that does not understand the new mobile messages will send an error message, and communications with the mobile node will be able to continue, albeit without the direct routing optimization.

How Mobile IPv6 Works

To implement Mobile IPv6, you need a home agent on the home subnet on which the mobile node's home address resides. The IPv6 home address (HA) is assigned to the mobile node. The mobile node obtains a new IPv6 address (the CoA) on networks to which it connects. The home agent accepts BUs from the mobile node informing the agent of the mobile node's location. The home agent then acts as proxy for the mobile node, intercepting traffic to the mobile node's home address and tunneling it to the mobile node.

The mobile node informs a home agent on its original home network about its new address, and the correspondent node communicates with the mobile node about the CoA. Because of the use of ingress filtering, the mobile node reverses tunnel return traffic to the home agent, so that the mobile node source address (that is, its home address) will always be topographically correct.

Mobile IPv6 is the ability of a mobile node to bypass the home agent when sending IP packets to a correspondent node. Optional extensions make direct routing possible in Mobile IPv6, though the extensions might not be implemented in all deployments of Mobile IPv6.

Direct routing is built into Mobile IPv6, and the direct routing function uses the IPv6 routing header and the IPv6 destination options header. The routing header is used for sending packets to the mobile node using its current CoA, and the new home address destination option is used to include the mobile node's home address, because the current CoA is the source address of the packet.

IPv6 NEMO

The NEMO basic support protocol enables mobile IPv6 networks to attach to different points in the Internet. This protocol is an extension of Mobile IPv6 and allows session continuity for every node in the mobile network as the network moves. NEMO also allows every node in the mobile network to be reachable while the user is moving. The mobile router, which connects the network to the Internet, runs the NEMO basic support protocol with its home agent (HA). NEMO allows network mobility to be transparent to the nodes inside the mobile network.

The NEMO router maintains a mobile route, which is the default route for IPv6 over the roaming interface.

Mobile IPv6 Home Agent

The home agent is one of three key components in Mobile IPv6. The home agent works with the correspondent node and mobile node to enable Mobile IPv6 functionality:

- Home agent—The home agent maintains an association between the mobile node's home IPv4 or IPv6 address and its CoA (loaned address) on the foreign network.
- Correspondent node—The correspondent node is the destination IPv4 or IPv6 host in session with a mobile node.
- Mobile node—An IPv4 or IPv6 host that maintains network connectivity using its home IPv4 or IPv6 address, regardless of the link (or network) to which it is connected.

The following sections describe Mobile IPv6 home agent functionality:

- [Binding Cache in Mobile IPv6 Home Agent, page 4](#)
- [Binding Update List in Mobile IPv6 Home Agent, page 4](#)
- [Home Agents List, page 4](#)

- [NEMO-Compliant Home Agent, page 4](#)

Binding Cache in Mobile IPv6 Home Agent

A separate binding cache is maintained by each IPv6 node for each of its IPv6 addresses. When the router sends a packet, it searches the binding cache for an IPv6 address before it searches the neighbor discovery conceptual destination cache.

The binding cache for any one of a node's IPv6 addresses may contain one entry for each mobile node home address. The contents of all of a node's binding cache entries are cleared when it reboots.

Binding cache entries are marked either as home registration or correspondent registration entries. A home registration entry is deleted when its binding lifetime expires; other entries may be replaced at any time through a local cache replacement policy.

Binding Update List in Mobile IPv6 Home Agent

A binding update (BU) list is maintained by each mobile node. The BU list records information for each BU sent by this mobile node whose lifetime has not yet expired. The BU list includes all BUs sent by the mobile node—those bindings sent to correspondent nodes, and those bindings sent to the mobile node's home agent.

The mobility extension header has a new routing header type and a new destination option, and it is used during the BU process. This header is used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings.

Home Agents List

A home agents list is maintained by each home agent and each mobile node. The home agents list records information about each home agent from which this node has recently received a router advertisement in which the home agent (H) bit is set.

Each home agent maintains a separate home agents list for each link on which it is serving as a home agent. This list is used by a home agent in the dynamic home agent address discovery mechanism. Each roaming mobile node also maintains a home agents list that enables it to notify a home agent on its previous link when it moves to a new link.

NEMO-Compliant Home Agent

Protocol extensions to Mobile IPv6 are used to enable support for network mobility. The extensions are backward-compatible with existing Mobile IPv6 functionality. A NEMO-compliant home agent can operate as a Mobile IPv6 home agent.

The dynamic home agent address discovery (DHAAD) mechanism allows a mobile node to discover the address of the home agent on its home link. The following list describes DHAAD functionality and features:

- The mobile router sends Internet Control Message Protocol (ICMP) home agent address discovery requests to the Mobile IPv6 home agent's anycast address for the home subnet prefix.
- A new flag (R) is introduced in the DHAAD request message, indicating the desire to discover home agents that support mobile routers. This flag is added to the DHAAD reply message as well.
- On receiving the home agent address discovery reply message, the mobile router discovers the home agents operating on the home link.

- The mobile router attempts home registration to each of the home agents until its registration is accepted. The mobile router waits for the recommended length of time between its home registration attempts with each of its home registration attempts.

Implicit Prefix Registration

When using implicit prefix registration, the mobile router does not register any prefixes as part of the binding update with its home agent. This function requires a static configuration at the home agent, and the home agent must have the information of the associated prefixes with the given mobile router for it to set up route forwarding.

Explicit Prefix Registration

When using explicit prefix registration, the mobile router presents a list of prefixes to the home agent as part of the binding update procedure. If the home agent determines that the mobile router is authorized to use these prefixes, it sends a bind acknowledgment message.

Packet Headers in Mobile IPv6

The basic IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). Fields were removed from the IPv6 header compared with the IPv4 header because, in IPv6, fragmentation is not handled by routers and checksums at the network layer are not used. Instead, fragmentation in IPv6 is handled by the source of a packet and checksums at the data link layer and transport layer are used. Additionally, the basic IPv6 packet header and options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

Mobile IPv6 uses the routing and destination option headers for communications between the mobile node and the correspondent node. The new mobility option header is used only for the BU process.

Several ICMP message types have been defined to support Mobile IPv6. IPv6 access lists can be configured to allow IPv6 access list entries matching Mobile-IPv6-specific ICMP messages to be configured and to allow the definition of entries to match packets containing Mobile IPv6 extension headers.

For further information on IPv6 packet headers, refer to the [Implementing IPv6 Addressing and Basic Connectivity](#) module.

IPv6 Neighbor Discovery with Mobile IPv6

The IPv6 neighbor discovery feature has the following modifications to allow the feature to work with Mobile IPv6:

- Modified router advertisement message format—has a single flag bit that indicates home agent service
- Modified prefix information option format—allows a router to advertise its global address
- New advertisement interval option format
- New home agent information option format
- Changes to sending router advertisements
- Provide timely movement detection for mobile nodes

IPv6 Neighbor Discovery Duplicate Address Detection in NEMO

IPv6 routers are required to run duplicate address detection (DAD) on all IPv6 addresses obtained in stateless and stateful autoconfiguration modes before assigning them to any of its interfaces. Whenever a mobile router roams and obtains an IPv6 address, the mobile router must perform DAD on the newly obtained care-of address and on its link-local address in order to avoid address collisions.

However, the DAD feature adds significant handoff delays in certain Layer 2 environments. These delays may be avoided by using optimistic DAD techniques. NEMO supports optimization options for omitting DAD on care-of address or on both the care-of address and link-local address.

For further information on IPv6 neighbor discovery, refer to the *Implementing IPv6 Addressing and Basic Connectivity* module.

Mobile IPv6 Tunnel Optimization

Mobile IPv6 tunnel optimization enables routing over a native IPv6 tunnel infrastructure, allowing Mobile IPv6 to use all IPv6 tunneling infrastructure features, such as Cisco Express Forwarding switching support.

After the home agent receives a valid BU request from a mobile node, it sets up its endpoint of the bidirectional tunnel. This process involves creating a logical interface with the encapsulation mode set to IPv6/IPv6, the tunnel source to the home agent's address on the mobile node's home link, and the tunnel destination set to the mobile node's registered care-of address. A route will be inserted into the routing table for the mobile node's home address via the tunnel.

IPv6 Host Group Configuration

Users can create mobile user or group policies using the IPv6 host group configuration. The host group profile lookup interface will allow the lookup of the profile associated with the sender of the BU using any of the search keys:

- Profile name
- IPv6 address
- Network address identifier (NAI)

The host profile lookup interface also specifies the authentication properties for the IPv6 mobile node by creating either a unidirectional or bidirectional security parameter index (SPI).

A group profile is activated after the SPI option is configured and either an NAI or an IPv6 address is configured. In addition, a profile is deactivated if the minimum required options are not configured. If any active profile that has active bindings gets deactivated or removed, all bindings associated to that profile are revoked.

Mobile IPv6 Node Identification Based on NAI

A mobile node can identify itself using its home address as an identifier. The Mobile IPv6 protocol messages use this identifier in their registration messages. However, for certain deployments it is essential that the mobile node has the capability to identify itself using a logical identifier, such as NAI, rather than a network address. The mobile node identifier option for Mobile IPv6 allows a mobile node to be identified by NAI rather than IPv6 address. This feature enables the network to give a dynamic IPv6

address to a mobile node and authenticate the mobile node using authentication, authorization, and accounting (AAA). This option should be used when either Internet Key Exchange (IKE) or IPsec is not used for protecting BUs or binding acknowledgments (BAs).

In order to provide roaming services, a standardized method, such as NAI or a mobile node home address, is needed for identifying users. Roaming may be loosely defined as the ability to use any one of multiple Internet service providers (ISPs) while maintaining a formal, customer-vendor relationship with only one. Examples of where roaming capabilities might be required include ISP confederations and ISP-provided corporate network access support. Other entities interested in roaming capability may include the following:

- Regional ISPs, operating within a particular state or province, that want to combine efforts with those of other regional providers to offer dialup service over a wider area.
- National ISPs that want to combine their operations with those of one or more ISPs in another country to offer more comprehensive dialup service in a group of countries or on a continent.
- Wireless LAN hot spots that provide service to one or more ISPs.
- Businesses that want to offer their employees a comprehensive package of dialup services on a global basis. Those services may include Internet access and secure access to corporate intranets using a VPN.

Authentication Protocol for Mobile IPv6

The authentication protocol for Mobile IPv6 support secures mobile node and home agent signaling using the MN-HA mobility message authentication option, which authenticates the BU and BA messages based on the shared-key-based security association between the mobile node (MN) and the HA. This feature allows Mobile IPv6 to be deployed in a production environment where a non-IPsec authentication method is required. MN-HA consists of a mobility SPI, a shared key, an authentication algorithm, and the mobility message replay protection option.

The mobility SPI is a number from 256 through 4,294,967,296. The key consists of an arbitrary value and is 16 octets in length. The authentication algorithm used is HMAC_SHA1. The replay protection mechanism may use either the sequence number option or the time-stamp option. The MN-HA mobility message authentication option must be the last option in a message with a mobility header if it is the only mobility message authentication option in the message.

When a BU or BA message is received without the MN-HA option and the entity receiving it is configured to use the MN-HA option or has the shared-key-based mobility security association for the mobility message authentication option, the entity discards the received message.

The mobility message replay protection option allows the home agent to verify that a BU has been freshly generated by the mobile node and not replayed by an attacker from some previous BU. This functionality is especially useful for cases where the home agent does not maintain stateful information about the mobile node after the binding entry has been removed. The home agent performs the replay protection check after the BU has been authenticated. The mobility message replay protection option is used by the mobile node for matching the BA with the BU. When the home agent receives the mobility message replay protection option in BU, it must include the mobility message replay protection option in the BA.

How to Implement Mobile IPv6

The following tasks explain how to enable and configure Mobile IPv6:

- [Enabling Mobile IPv6 on the Router, page 8](#)

- [Configuring Binding Information for Mobile IPv6, page 9](#)
- [Enabling and Configuring NEMO on the IPv6 Mobile Router, page 10](#)
- [Enabling NEMO on the IPv6 Mobile Router Home Agent, page 12](#)
- [Enabling Roaming on the IPv6 Mobile Router Interface, page 13](#)
- [Filtering Mobile IPv6 Protocol Headers and Options, page 14](#)
- [Controlling ICMP Unreachable Messages, page 15](#)
- [Verifying Native IPv6 Tunneling for Mobile IPv6, page 16](#)
- [Configuring and Verifying Host Groups for Mobile IPv6, page 17](#)
- [Customizing Mobile IPv6 on the Interface, page 19](#)
- [Monitoring and Maintaining Mobile IPv6 on the Router, page 20](#)

Enabling Mobile IPv6 on the Router

The following task describes how to enable Mobile IPv6 on a specified interface and display Mobile IPv6 information. You can customize interface configuration parameters before you start Mobile IPv6 (see the “[Customizing Mobile IPv6 on the Interface](#)” section on page 19) or while Mobile IPv6 is in operation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mobile home-agent** [*preference preference-value*]
5. **exit**
6. **exit**
7. **show ipv6 mobile globals**
8. **show ipv6 mobile home-agent** [*interface-type interface-number* [*prefix*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 2	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 mobile home-agent [preference <i>preference-value</i>] Example: Router(config-if)# ipv6 mobile home-agent	Initializes and starts the Mobile IPv6 home agent on a specific interface.
Step 5	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 6	exit Example: Router(config)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 7	show ipv6 mobile globals Example: Router# show ipv6 mobile globals	Displays global Mobile IPv6 parameters.
Step 8	show ipv6 mobile home-agent [<i>interface-type</i> <i>interface-number</i> [<i>prefix</i>]] Example: Router# show ipv6 mobile home-agent	Displays local and discovered neighboring home agents.

Configuring Binding Information for Mobile IPv6

Before you start Mobile IPv6 on a specified interface, you can configure binding information on the router. The following task describes how to configure and verify binding information on the IPv6 router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mobile home-agent**
4. **binding** [*access access-list-name* | *auth-option* | *seconds* | *maximum* | *refresh*]
5. **exit**
6. **exit**
7. **show ipv6 mobile binding** [*care-of-address address* | **home-address** *address* | *interface-type interface-number*]
8. **show ipv6 mobile traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mobile home-agent Example: Router(config)# ipv6 mobile home-agent	Places the router in home-agent configuration mode.
Step 4	binding [access <i>access-list-name</i> <i>auth-option</i> <i>seconds</i> <i>maximum</i> <i>refresh</i>] Example: Router(config-ha)# binding	Configures binding options for the Mobile IPv6 home agent feature.
Step 5	exit Example: Router(config-ha)# exit	Exits home-agent configuration mode, and returns the router to global configuration mode.
Step 6	exit Example: Router(config)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 7	show ipv6 mobile binding [care-of-address <i>address</i> home-address <i>address</i> <i>interface-type</i> <i>interface-number</i>] Example: Router# show ipv6 mobile binding	Displays information about the binding cache.
Step 8	show ipv6 mobile traffic Example: Router# show ipv6 mobile traffic	Displays information about BUs received and BAs sent.

Enabling and Configuring NEMO on the IPv6 Mobile Router

The NEMO basic support protocol enables mobile IPv6 networks to attach to different points in the Internet. This task describes how to enable and configure NEMO on the IPv6 mobile router, and how to verify NEMO configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mobile router**
4. **home-network** *ipv6-prefix* [**discover**] [**priority** *priority-number*]
5. **home-address** **home-network** {*ipv6-address-identifier* | **eui-64**}
6. **explicit-prefix**
7. **register** {**extend** **expire** *seconds* **retry** *number* **interval** *seconds* | **lifetime** *seconds* | **retransmit** *initial milliseconds* **maximum** *milliseconds* **retry** *number*}
8. **exit**
9. **exit**
10. **show ipv6 mobile router** [**running-config** | **status**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mobile router Example: Router(config)# ipv6 mobile router	Enables IPv6 NEMO functionality on a router, and places the router in IPv6 mobile router configuration mode.
Step 4	home-network <i>ipv6-prefix</i> [discover] [priority <i>priority-number</i>] Example: Router(IPv6-mobile-router)# home-network 2001::/32	Specifies the home network's IPv6 prefix on the mobile router.
Step 5	home-address home-network { <i>ipv6-address-identifier</i> eui-64 } Example: Router(IPv6-mobile-router)# home-address home-network eui-64	Specifies the home address using the home network or interface identifier.
Step 6	explicit-prefix Example: Router(IPv6-mobile-router)# explicit-prefix	Registers IPv6 prefixes connected to the IPv6 mobile router.

	Command or Action	Purpose
Step 7	register { extend expire seconds retry number interval seconds lifetime seconds retransmit initial milliseconds maximum milliseconds retry number } Example: Router(IPv6-mobile-router)# register lifetime 600	Controls the registration parameters of the IPv6 mobile router.
Step 8	exit Example: Router(IPv6-mobile-router)# exit	Exits IPv6 mobile router configuration mode, and returns the router to global configuration mode.
Step 9	exit Example: Router(config)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 10	show ipv6 mobile router [running-config status] Example: Router# show ipv6 mobile router	Displays configuration information and monitoring statistics about the IPv6 mobile router.

Enabling NEMO on the IPv6 Mobile Router Home Agent

This task describes how to enable and configure NEMO on the IPv6 mobile router home agent.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router nemo**
4. **distance** [*mobile-distance*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ipv6 router nemo Example: Router(config)# ipv6 router nemo	Enables the NEMO routing process on the home agent and place the router in router configuration mode.
Step 4	distance [<i>mobile-distance</i>] Example: Router(config-rtr)# distance 10	Defines an administrative distance for NEMO routes.

Enabling Roaming on the IPv6 Mobile Router Interface

This task describes how to enable roaming on the IPv6 mobile router interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mobile router-service roam** [**bandwidth-efficient** | **cost-efficient** | **optimistic** | **semi-optimistic** | **priority value**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 mobile router-service roam [bandwidth-efficient cost-efficient optimistic semi-optimistic priority value] Example: Router(config-if)# ipv6 mobile router-service roam	Enables the IPv6 mobile router interface to roam.

Filtering Mobile IPv6 Protocol Headers and Options

IPv6 extension headers have been developed to support the use of option headers specific to Mobile IPv6. The IPv6 mobility header, the type 2 routing header, and the destination option header allow the configuration of IPv6 access list entries that match Mobile-IPv6-specific ICMPv6 messages and allow the definition of entries to match packets that contain the new and modified IPv6 extension headers.

This task describes how to enable filtering of Mobile IPv6 protocol headers and options. For more information on how to create, configure, and apply IPv6 access lists, refer to the [Implementing Traffic Filters and Firewalls for IPv6 Security](#) module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name*
4. **permit icmp** { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
or
deny icmp { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	<pre>ipv6 access-list access-list-name</pre> <p>Example: Router(config)# ipv6 access-list list1</p>	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.
Step 4	<pre>permit icmp {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [operator [port-number]] [icmp-type [icmp-code] icmp-message] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [routing] [routing-type routing-number] [sequence value] [time-range name]</pre> <p>or</p> <pre>deny icmp {source-ipv6-prefix/prefix-length any host source-ipv6-address auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth} [operator [port-number]] [icmp-type [icmp-code] icmp-message] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [routing] [routing-type routing-number] [sequence value] [time-range name]</pre> <p>Example: Router(config-ipv6-acl)# permit icmp host 2001:0DB8:0:4::32 any routing-type 2</p> <p>or</p> <pre>Router(config-ipv6-acl)# deny icmp host 2001:0DB8:0:4::32 any routing-type 2</pre>	<p>Specifies permit or deny conditions for Mobile-IPv6-specific option headers in an IPv6 access list.</p> <ul style="list-style-type: none"> The <i>icmp-type</i> argument can be (but is not limited to) one of the following Mobile-IPv6-specific options: <ul style="list-style-type: none"> dhaad-request—numeric value is 144 dhaad-reply—numeric value is 145 mpd-solicitation—numeric value is 146 mpd-advertisement—numeric value is 147 When the dest-option-type keyword with the <i>doh-number</i> or <i>doh-type</i> argument is used, IPv6 packets are matched against the destination option extension header within each IPv6 packet header. When the mobility keyword is used, IPv6 packets are matched against the mobility extension header within each IPv6 packet header. When the mobility-type keyword with the <i>mh-number</i> or <i>mh-type</i> argument is used, IPv6 packets are matched against the mobility-type option extension header within each IPv6 packet header. When the routing-type keyword and <i>routing-number</i> argument are used, IPv6 packets are matched against the routing-type option extension header within each IPv6 packet header.

Controlling ICMP Unreachable Messages

When IPv6 is unable to route a packet, it generates an appropriate ICMP unreachable message directed toward the source of the packet. This task describes how to control ICMP unreachable messages for any packets arriving on a specified interface.

SUMMARY STEPS

1. enable
2. configure terminal
3. interface *type number*
4. ipv6 unreachable

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 unreachable Example: Router(config-if)# ipv6 unreachable	Enables the generation of ICMPv6 unreachable messages for any packets arriving on the specified interface.

Verifying Native IPv6 Tunneling for Mobile IPv6

The following task shows how to verify IPv6 tunnel information in Mobile IPv6.

Using the native IPv6 tunneling (or generic routing encapsulation [GRE]) infrastructure improves the scalability and switching performance of the home agent. After the home agent sends a BU from a mobile node, a tunnel interface is created with the encapsulation mode set to IPv6/IPv6, the source address set to that of the home agent address on the home interface of the mobile node, and the tunnel destination set to that of the CoA of the mobile node.

These features are transparent and need not be configured in order to work with Mobile IPv6. For further information on IPv6 tunneling and how to implement GRE tunneling in IPv6, see the *Implementing Tunneling for IPv6* module.

SUMMARY STEPS

1. **enable**
2. **show ipv6 mobile tunnels** [**summary** | **tunnel** *if-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ipv6 mobile tunnels [summary tunnel if-number] Example: Router# show ipv6 mobile tunnels	Lists the Mobile IPv6 tunnels on the home agent.

Configuring and Verifying Host Groups for Mobile IPv6

This task describes how to configure and verify host group information for Mobile IPv6.

Users can create mobile user or group policies using the host group configuration. The host group profile lookup interface will allow the lookup of the profile associated with the sender of the BU using the sender's profile name, IPv6 address, or NAI. The host profile lookup interface also specifies the authentication properties for the IPv6 mobile node by creating either a unidirectional or bidirectional SPI.

A mobile node can identify itself using its profile name or home address as an identifier, which the Mobile IPv6 protocol messages use as an identifier in their registration messages. However, for certain deployments it is essential that the mobile node has the capability to identify itself using a logical identifier such as NAI rather than a network address.

Restrictions

- You cannot configure two host group profiles with the same IPv6 address when using the IPv6 address option.
- You cannot configure a profile with the NAI option set to a realm name and the address option set to a specific IPv6 address. You can either remove the NAI option or specify a fully qualified user name for the NAI option.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 mobile home-agent**
- binding** [**access** *access-list-name* | **auth-option** | *seconds* | *maximum* | *refresh*]
- host group** *profile-name*
- address** { *ipv6-address* | **autoconfig** }
- nai** [**realm** | **user** | **macaddress**] { *user@realm* | *@realm* }

8. **authentication** {**inbound-spi** {*hex-in* | **decimal** *decimal-in*} **outbound-spi** {*hex-out* | **decimal** *decimal-out*} | **spi** {*hex-value* | **decimal** *decimal-value*}} **key** {*ascii string* | **hex string**} [**algorithm** *algorithm-type*] [**replay within** *seconds*]
9. **exit**
10. **exit**
11. **show ipv6 mobile host groups** [*profile-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mobile home-agent Example: Router(config)# ipv6 mobile home-agent	Places the router in home-agent configuration mode.
Step 4	binding [access <i>access-list-name</i> <i>auth-option</i> <i>seconds</i> <i>maximum</i> <i>refresh</i>] Example: Router(config-ha)# binding 15	Configures binding options for the Mobile IPv6 home agent feature.
Step 5	host group <i>profile-name</i> Example: Router(config-ha)# host group profile1	Creates a host configuration in Mobile IPv6. <ul style="list-style-type: none">• Multiple instances with different profile names can be created and used.
Step 6	address { <i>ipv6-address</i> autoconfig } Example: Router(config-ha)# address baba::1	Specifies the home address of the IPv6 mobile node.
Step 7	nai [realm user macaddress] { <i>user@realm</i> <i>@realm</i> } Example: Router(config-ha)# nai @cisco.com	Specifies the NAI for the IPv6 mobile node.

	Command or Action	Purpose
Step 8	authentication { inbound-spi { <i>hex-in</i> decimal <i>decimal-in</i> } outbound-spi { <i>hex-out</i> decimal <i>decimal-out</i> } spi { <i>hex-value</i> decimal <i>decimal-value</i> }} key { <i>ascii string</i> hex <i>string</i> } [algorithm <i>algorithm-type</i>] [replay <i>within seconds</i>] Example: Router(config-ha)# authentication spi 500 key ascii cisco	Specifies the authentication properties for the IPv6 mobile node by creating either a unidirectional or bidirectional SPI.
Step 9	exit Example: Router(config-ha)# exit	Exits home-agent configuration mode, and returns the router to global configuration mode.
Step 10	exit Example: Router(config)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 11	show ipv6 mobile host groups [<i>profile-name</i>] Example: Router# show ipv6 mobile host groups	Displays information about Mobile IPv6 host groups.

Customizing Mobile IPv6 on the Interface

This task describes several ways to customize interface configuration parameters for your router configuration, such as:

- Configuring an advertisement interval option to be sent in router advertisements (RAs)
- Configuring which IPv6 prefixes are included in IPv6 RAs
- Configuring the interval between IPv6 RA transmissions on an interface

You can set these interface configuration parameters before you start Mobile IPv6 or while Mobile IPv6 is in operation. You can customize any of these parameters, as desired.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mobile home-agent** [*preference preference-value*]
5. **ipv6 nd advertisement-interval**
6. **ipv6 nd prefix** {*ipv6-prefix/prefix-length* | **default**} [[*valid-lifetime preferred-lifetime* | **at valid-date preferred-date**] | **infinite** | **no-advertise** | **off-link** | **no-rtr-address** | **no-autoconfig**]
7. **ipv6 nd ra interval** {*maximum-secs* [*minimum-secs*] | **msec** *maximum-msecs* [*minimum-msecs*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface serial 3	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 mobile home-agent [preference <i>preference-value</i>] Example: Router(config-if)# ipv6 mobile home-agent preference 10	Configures the Mobile IPv6 home agent preference value on the interface.
Step 5	ipv6 nd advertisement-interval Example: Router(config-if)# ipv6 nd advertisement-interval	Configures the advertisement interval option to be sent in RAs.
Step 6	ipv6 nd prefix { <i>ipv6-prefix/prefix-length</i> default } [[<i>valid-lifetime preferred-lifetime</i> at <i>valid-date preferred-date</i>] infinite no-advertise off-link no-rtr-address no-autoconfig] Example: Router(config-if)# ipv6 nd prefix 2001:0DB8::/35 1000 900	Configures which IPv6 prefixes are included in IPv6 RAs.
Step 7	ipv6 nd ra interval { <i>maximum-secs</i> [<i>minimum-secs</i>] msec <i>maximum-msecs</i> [<i>minimum-msecs</i>]} Example: Router(config-if)# ipv6 nd ra interval 201	Configures the interval between IPv6 RA transmissions on an interface.

Monitoring and Maintaining Mobile IPv6 on the Router

This task describes many ways to monitor and maintain a Mobile IPv6 router, such as:

- Clear the Mobile IPv6 binding cache on a router
- Clear the neighboring home agents list

- Clear counters associated with Mobile IPv6
- Enable display of debugging information for Mobile IPv6
- Display memory used by the Mobile IPv6 internal structure

This task is optional, and these commands are used only as necessary to clear or display Mobile IPv6 information or enable Mobile IPv6 debugging.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 mobile binding** [*care-of-address prefix* | *home-address prefix* | *interface-type interface-number*]
3. **clear ipv6 mobile home-agents** [*interface-type interface-number*]
4. **clear ipv6 mobile traffic**
5. **debug ipv6 mobile** {*binding-cache* | *forwarding* | *home-agent* | *registration*}
6. **debug ipv6 mobile networks**
7. **debug ipv6 mobile router** [*detail*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear ipv6 mobile binding [<i>care-of-address prefix</i> <i>home-address prefix</i> <i>interface-type interface-number</i>] Example: Router# clear ipv6 mobile binding	Clears the Mobile IPv6 binding cache on a router.
Step 3	clear ipv6 mobile home-agents [<i>interface-type interface-number</i>] Example: Router# clear ipv6 mobile home-agents	Clears the neighboring home agents list.
Step 4	clear ipv6 mobile traffic Example: Router# clear ipv6 mobile traffic	Clears the counters associated with Mobile IPv6.
Step 5	debug ipv6 mobile { <i>binding-cache</i> <i>forwarding</i> <i>home-agent</i> <i>registration</i> } Example: Router# debug ipv6 mobile registration	Enables the display of debugging information for Mobile IPv6.

	Command or Action	Purpose
Step 6	debug ipv6 mobile networks Example: Router# debug ipv6 mobile networks	Displays debugging messages for IPv6 mobile networks.
Step 7	debug ipv6 mobile router [detail] Example: Router# debug ipv6 mobile router	Displays debugging messages for the IPv6 mobile router.

Examples

The following sections provides sample **show** command output to verify Mobile IPv6 configuration and operation:

- [Sample Output from the show ipv6 mobile binding Command, page 22](#)
- [Sample Output from the show ipv6 mobile globals Command, page 22](#)
- [Sample Output from the show ipv6 mobile home-agent Command, page 23](#)
- [Sample Output from the show ipv6 mobile host groups Command, page 23](#)
- [Sample Output from the show ipv6 mobile router Command, page 23](#)
- [Sample Output from the show ipv6 mobile traffic Command, page 23](#)
- [Sample Output from the show ipv6 mobile tunnels Command, page 24](#)

Sample Output from the show ipv6 mobile binding Command

The following example displays information about the binding cache:

```
Router # show ipv6 mobile binding

Mobile IPv6 Binding Cache Entries:
2001:DB8:2000::1111
via care-of address 2001:DB8::A8BB:CCFF:FE01:F611
home-agent 2001:DB8:2000::2001
Prefix 2001:DB8:8000::/64
Prefix 2001:DB8:2000::1111/128
Prefix 2001:DB8:1000::1111/128 installed
state ACTIVE, sequence 23, flags AHRlK
lifetime: remaining 44 (secs), granted 60 (secs), requested 60 (secs)
interface Ethernet0/2
tunnel interface Tunnel0
0 tunneled, 0 reversed tunneled
Selection matched 1 bindings
```

Sample Output from the show ipv6 mobile globals Command

In the following example, the **show ipv6 mobile globals** command displays the binding parameters:

```
Router# show ipv6 mobile globals

Mobile IPv6 Global Settings:

1 Home Agent service on following interfaces:
  Ethernet1/2
Bindings:
  Maximum number is unlimited.
```

```

1 bindings are in use
1 bindings peak
Binding lifetime permitted is 262140 seconds
Recommended refresh time is 300 seconds

```

Sample Output from the show ipv6 mobile home-agent Command

In the following example, the fact that no neighboring mobile home agents were found is displayed:

```

Router# show ipv6 mobile home-agent

Home Agent information for Ethernet1/3
Configured:
  FE80::20B:BFFF:FE33:501F
  preference 0 lifetime 1800
  global address 2001:0DB8:1::2/64
Discovered Home Agents:
  FE80::4, last update 0 min
  preference 0 lifetime 1800
  global address 2001:0DB8:1::4/64

```

Sample Output from the show ipv6 mobile host groups Command

In the following example, information about a host group named localhost is displayed:

```

Router# show ipv6 mobile host groups

Mobile IPv6 Host Configuration
Mobile Host List:

Host Group Name: localhost
NAI: sai@cisco.com
Address: CAB:C0:CA5A:CA5A::CA5A

Security Association Entry:
  SPI: (Hex: 501) (Decimal Int: 1281)
  Key Format: Hex      Key: baba
  Algorithm: HMAC_SHA1
  Replay Protection: On      Replay Window: 6 secs

```

Sample Output from the show ipv6 mobile router Command

The following example provides information about the IPv6 mobile router status when the router configured with IPv6 NEMO:

```

Router# show ipv6 mobile router

Mobile Reverse Tunnel established
-----
using Nemo Basic mode
Home Agent: 2001:DB8:2000::2001
CareOf Address: 2001:DB8::A8BB:CCFF:FE01:F611
Attachment Router: FE80::A8BB:CCFF:FE01:F511
Attachment Interface: Ethernet1/1
Home Network: 2001:DB8:2000:0:FDFD:FFFF:FFFF:FFFE/64
Home Address: 2001:DB8:2000::1111

```

Sample Output from the show ipv6 mobile traffic Command

In the following example, information about Mobile IPv6 traffic is displayed:

```

Router# show ipv6 mobile traffic

MIPv6 statistics:
  Rcvd: 6477 total
      0 truncated, 0 format errors

```

```

    0 checksum errors
    Binding Updates received:6477
    0 no HA option, 0 BU's length
    0 options' length, 0 invalid CoA
    Sent: 6477 generated
    Binding Acknowledgements sent:6477
    6477 accepted (0 prefix discovery required)
    0 reason unspecified, 0 admin prohibited
    0 insufficient resources, 0 home reg not supported
    0 not home subnet, 0 not home agent for node
    0 DAD failed, 0 sequence number
    Binding Errors sent:0
    0 no binding, 0 unknown MH

Home Agent Traffic:
    6477 registrations, 0 deregistrations
    00:00:23 since last accepted HA registration
    unknown time since last failed HA registration
    unknown last failed registration code
    Traffic forwarded:
    0 tunneled, 0 reversed tunneled
    Dynamic Home Agent Address Discovery:
    1 requests received, 1 replies sent
    Mobile Prefix Discovery:
    0 solicitations received, 0 advertisements sent

```

Sample Output from the show ipv6 mobile tunnels Command

The following example displays information about the Mobile IPv6 tunnels on the home agent:

```
Router# show ipv6 mobile tunnels
```

```

Tunnel1:
Source: 2001:0DB1:1:1
Destination: 2001:0DB1:2:1
Encapsulation Mode: IPv6/IPv6
Egress Interface: Ethernet 1/0
Switching Mode: Process
Keep-Alive: Not Supported
Path MTU Discovery: Enabled
Input: 20 packets, 1200 bytes, 0 drops
Output: 20 packets, 1200 bytes, 0 drops
NEMO Options: Not Supported

```

Configuration Examples for Implementing Mobile IPv6

This section provides the following configuration examples:

- [Enabling Mobile IPv6 on the Router: Example, page 25](#)
- [Enabling and Configuring NEMO on the IPv6 Mobile Router: Example, page 25](#)
- [Enabling NEMO on the IPv6 Mobile Router Home Agent: Example, page 26](#)
- [Enabling Roaming on the IPv6 Mobile Router Interface: Example, page 27](#)
- [Configuring Host Groups for Mobile IPv6: Example, page 27](#)

Enabling Mobile IPv6 on the Router: Example

The following example shows how to configure and enable Mobile IPv6 on a specified interface:

```
Router> enable
Router# config terminal
Router(config)# interface Ethernet 1
Router(config-if)# ipv6 mobile home-agent
```

Enabling and Configuring NEMO on the IPv6 Mobile Router: Example

The following example shows how to enable and configure NEMO on the IPv6 mobile router. The /128 subnet must be used; otherwise, the IPv6 mobile router will fail to register because it will believe the home network is locally connected:

```
ipv6 unicast-routing
!
interface Loopback0
no ip address
ipv6 address 2001:0DB8:2000::1111/128
ipv6 nd ra mtu suppress
!
interface Loopback1
no ip address
ipv6 address 2001:0DB8:1000::1111/128
ipv6 nd ra mtu suppress
!
interface Ethernet0/0
description Roaming Interface to AR2
no ip address
ipv6 address autoconfig
ipv6 enable
ipv6 nd ns-interval 5000
ipv6 mobile router-service roam
ipv6 rip home enable
!
interface Ethernet0/1
description Mobile Network Interface
no ip address
ipv6 address 2001:0DB8:8000::8001/64
ipv6 enable
ipv6 nd advertisement-interval
ipv6 nd ra interval msec 1000
ipv6 rip home enable
!
interface Ethernet1/1
description Roaming Interface to AR1
no ip address
ipv6 address autoconfig
ipv6 enable
ipv6 nd ns-interval 5000
ipv6 mobile router-service roam priority 99
ipv6 rip home enable
!
ipv6 router rip home
!
ipv6 mobile router
host group mr-host-group
nai mrl@cisco.com
address 2001:0DB8:2000::2001
authentication spi hex 100 key ascii hi
```

```

exit
home-network 2001:0DB8:2000::/64 discover priority 127
home-network 2001:0DB8:1000::/64 discover
home-address home-network ::1111
explicit-prefix
register lifetime 60
register retransmit initial 1000 maximum 1000 retry 1
register extend expire 20 retry 1 interval 1

```

Enabling NEMO on the IPv6 Mobile Router Home Agent: Example

The following example shows how to enable and configure NEMO on the IPv6 mobile router home agent. The anycast address is needed for DHAAD to work. The **redistribute nemo** command redistributes NEMO routes into the routing protocol:

```

ipv6 unicast-routing
!
interface Ethernet0/2
description To Network
no ip address
no ipv6 address
ipv6 address 2001:0DB8:2000::2001/64
ipv6 address 2001:0DB8:2000::FDF5:FFFF:FFFF:FFFE/64 anycast
ipv6 enable
ipv6 nd advertisement-interval
ipv6 nd ra lifetime 2
ipv6 nd ra interval msec 1000
ipv6 mobile home-agent preference 100
ipv6 mobile home-agent
ipv6 rip home enable
!
interface Ethernet2/2
description To CN2
no ip address
no ipv6 address
ipv6 address 2001:0DB8:3000::3001/64
ipv6 enable
ipv6 rip home enable
!
ipv6 router nemo
!
ipv6 router rip home
redistribute nemo
poison-reverse
!
ipv6 mobile home-agent
host group mr-host-group
nai mr1@cisco.com
address 2001:0DB8:2000::1111
authentication spi hex 100 key ascii hi
exit
host group mr2-host-group
nai mr2@cisco.com
address 2001:0DB8:2000::2222
authentication spi decimal 512 key hex 12345678123456781234567812345678
exit

```


Enabling Roaming on the IPv6 Mobile Router Interface: Example

The following example shows how to enable roaming on the IPv6 mobile router interface:

```
Router(config)# interface ethernet 0/0
Router(config-if)# ipv6 mobile router-service roam
```

Configuring Host Groups for Mobile IPv6: Example

The following example shows how to configure a Mobile IPv6 host group named group1:

```
ipv6 mobile host group group1
  nai sri@cisco.com
  address autoconfig
  authentication spi 500 key ascii cisco
```

Additional References

The following sections provide references related to the Implementing Mobile IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	Start Here: Cisco IOS Software Release Specifics for IPv6 Features
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IPv6 simplified packet headers, IPv6 neighbor discovery, IPv6 stateless autoconfiguration, IPv6 stateful autoconfiguration	“ Implementing IPv6 Addressing and Basic Connectivity ” module of the <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 access lists	“ Implementing Traffic Filters and Firewalls for IPv6 Security ” module of the <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 tunneling	“ Implementing Tunneling for IPv6 ” module of the <i>Cisco IOS IPv6 Configuration Guide</i>
IPv4 mobility configuration and commands	<ul style="list-style-type: none"> Cisco IOS IP Mobility Configuration Guide Cisco IOS IP Mobility Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 3775	<i>Mobility Support in IPv6</i>
RFC 3846	<i>Mobile IPv4 Extension for Carrying Network Access Identifiers</i>
RFC 3963	<i>Network Mobility (NEMO) Basic Support Protocol</i>
RFC 4282	<i>The Network Access Identifier</i>
RFC 4283	<i>Mobile Node Identifier Option for Mobile IPv6 (MIPv6)</i>
RFC 4285	<i>Authentication Protocol for Mobile IPv6</i>
draft-ietf-nemo-terminology	<i>Network Mobility Support Terminology</i>
draft-ietf-nemo-home-network-models	<i>NEMO Home Network Models</i>
draft-thubert-nemo-ipv4-traversal	<i>IPv4 Traversal for MIPv6 Mobile Routers</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about

all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **address (Mobile IPv6)**
- **authentication (Mobile IPv6)**
- **binding**
- **clear ipv6 mobile binding**
- **clear ipv6 mobile home-agents**
- **clear ipv6 mobile traffic**
- **debug ipv6 mobile**
- **debug ipv6 mobile networks**
- **debug ipv6 mobile router**
- **distance (IPv6 Mobile)**
- **eui-interface**
- **explicit-prefix**
- **home-address home-network**
- **home-network**
- **host group**
- **ipv6 mobile home-agent (global configuration)**
- **ipv6 mobile home-agent (interface configuration)**
- **ipv6 mobile router**
- **ipv6 mobile router-service roam**
- **ipv6 nd advertisement-interval**
- **ipv6 nd prefix**
- **ipv6 nd ra interval**
- **ipv6 router nemo**
- **ipv6 unreachable**
- **nai**
- **register (mobile router)**
- **show ipv6 mobile binding**
- **show ipv6 mobile globals**
- **show ipv6 mobile home-agent**
- **show ipv6 mobile host groups**
- **show ipv6 mobile router**
- **show ipv6 mobile traffic**
- **show ipv6 mobile tunnels**

Feature Information for Implementing Mobile IPv6

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(14)T or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing Mobile IPv6

Feature Name	Releases	Feature Information
Mobile IPv6 Home Agent	12.3(14)T 12.4	<p>The Mobile IPv6 feature uses the IPv6 address space to enable Mobile IP deployment in any kind of large environment. No foreign agent is needed to use Mobile IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Mobile IPv6 Home Agent, page 3 • Enabling Mobile IPv6 on the Router, page 8 • Configuring Binding Information for Mobile IPv6, page 9 • Customizing Mobile IPv6 on the Interface, page 19 • Enabling Mobile IPv6 on the Router: Example, page 25
IPv6 ACL Extensions for Mobile IPv6	12.4(2)T 12.2(33)SRB 12.2(33)SXI	<p>IPv6 access lists can be configured to allow IPv6 access list entries matching Mobile-IPv6-specific ICMP messages to be configured and to allow the definition of entries to match packets containing Mobile IPv6 extension headers.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Packet Headers in Mobile IPv6, page 5 • Filtering Mobile IPv6 Protocol Headers and Options, page 14 • Controlling ICMP Unreachable Messages, page 15

Table 1 *Feature Information for Implementing Mobile IPv6 (continued)*

Feature Name	Releases	Feature Information
Mobile IP—Mobile IPv6 HA phase 2	12.4(11)T	<p>This phase of development for Mobile IPv6 includes support for NAI, alternate authentication, and native IPv6 tunnel infrastructure.</p> <p>The following sections provide information about these features:</p> <ul style="list-style-type: none"> • Mobile IPv6 Tunnel Optimization, page 6 • IPv6 Host Group Configuration, page 6 • Mobile IPv6 Node Identification Based on NAI, page 6 • Authentication Protocol for Mobile IPv6, page 7 • Verifying Native IPv6 Tunneling for Mobile IPv6, page 16 • Configuring and Verifying Host Groups for Mobile IPv6, page 17 • Configuring Host Groups for Mobile IPv6: Example, page 27
Mobile Networks v6—Basic NEMO	12.4(20)T	<p>The network mobility (NEMO) basic support protocol enables mobile IPv6 networks to attach to different points in the Internet.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 NEMO, page 3 • NEMO-Compliant Home Agent, page 4 • IPv6 Neighbor Discovery Duplicate Address Detection in NEMO, page 6 • Enabling and Configuring NEMO on the IPv6 Mobile Router, page 10 • Enabling NEMO on the IPv6 Mobile Router Home Agent, page 12 • Enabling Roaming on the IPv6 Mobile Router Interface, page 13 • Enabling and Configuring NEMO on the IPv6 Mobile Router: Example, page 25 • Enabling NEMO on the IPv6 Mobile Router Home Agent: Example, page 26 • Enabling Roaming on the IPv6 Mobile Router Interface: Example, page 27

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 Multicast

First Published: July 28, 2003

Last Updated: February 27, 2009

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IPv6 multicast provides a third scheme, allowing a host to send a single data stream to a subset of all hosts (group transmission) simultaneously.

This module describes the concepts and tasks you need to implement IPv6 multicast on your network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPv6 Multicast” section on page 78](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IPv6 Multicast, page 2](#)
- [Restrictions for Implementing IPv6 Multicast, page 2](#)
- [Information About Implementing IPv6 Multicast, page 3](#)
- [How to Implement IPv6 Multicast, page 18](#)
- [Configuration Examples for Implementing IPv6 Multicast, page 69](#)
- [Additional References, page 72](#)
- [Command Reference, page 74](#)
- [Feature Information for Implementing IPv6 Multicast, page 78](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing IPv6 Multicast

- In order to enable IPv6 multicast routing on a router, you must first enable IPv6 unicast routing on the router. For information on how to enable IPv6 unicast routing on a router, refer to [Implementing IPv6 Addressing and Basic Connectivity](#).
- You must enable IPv6 unicast routing on all interfaces.
- This module assumes that you are familiar with IPv6 addressing and basic configuration. Refer to the [Implementing IPv6 Addressing and Basic Connectivity](#) module for more information.
- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Additional References](#)” section for IPv4 configuration and command reference information, as needed.

Restrictions for Implementing IPv6 Multicast

- IPv6 multicast for Cisco IOS software uses MLD version 2. This version of MLD is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a router running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- IPv6 multicast is supported only over IPv4 tunnels in Cisco IOS Release 12.3(2)T, Cisco IOS Release 12.2(18)S, and Cisco IOS Release 12.0(26)S.
- When using bidirectional (bidir) range in a network, all routers in that network must be able to understand the bidirectional range in the bootstrap message (BSM).
- IPv6 multicast routing is disabled by default when the **ipv6 unicast-routing** command is configured. On Cisco Catalyst 6500 and Cisco 7600 series routers, the **ipv6 multicast-routing** also must be enabled in order to use IPv6 unicast routing.

Platform-Specific Information and Restrictions

In Cisco IOS Release 12.0(26)S, IPv6 Multicast is supported on the Cisco 12000 series Internet router only on the following line cards:

- IP Service Engine (ISE):
 - 4-port Gigabit Ethernet ISE
 - 4-port OC-3c/STM-1c POS/SDH ISE
 - 8-port OC-3c/STM-1c POS/SDH ISE
 - 16-port OC-3c/STM-1c POS/SDH ISE
 - 4-port OC-12c/STM-4c POS/SDH ISE
 - 1-port OC-48c/STM-16c POS/SDH ISE
- Engine 4 Plus (E4+) Packet-over-SONET (POS):
 - 4-port OC-48c/STM-16c POS/SDH
 - 1-port OC-192c/STM-64c POS/SDH

On Cisco 12000 series line cards, the IPv6 multicast feature includes support for Protocol Independent Multicast sparse mode (PIM-SM), Multicast Listener Discovery (MLDv2), static mroutes, and the IPv6 distributed Multicast Forwarding Information Base (MFIB).

Forwarding of IPv6 multicast traffic is hardware-based on Cisco 12000 series IP Service Engine (ISE) line cards that support IPv6 multicast and software-based on all other supported Cisco 12000 series line cards.

On Cisco 12000 series ISE line cards, IPv6 multicast is implemented so that if the number of IPv6 multicast routes exceeds the hardware capacity of the ternary content addressable memory (TCAM), the following error message is displayed to describe how to increase the TCAM hardware capacity for IPv6 multicast routes:

```
EE48-3-IPV6_TCAM_CAPACITY_EXCEEDED: IPv6 multicast pkts will be software switched.  
To support more IPv6 multicast routes in hardware:  
Get current TCAM usage with: show controllers ISE <slot> tcam  
In config mode, reallocate TCAM regions e.g. reallocate Netflow TCAM to IPv6 Mcast  
hw-module slot <num> tcam carve rx_ipv6_mcast <v6-mcast-percent>  
hw-module slot <num> tcam carve rx_top_nf <nf-percent>  
Verify with show command that sum of all TCAM regions = 100%  
Reload the linecard for the new TCAM carve config to take effect  
WARNING: Recarve may affect other input features (ACL, CAR, MQC, Netflow)
```

TCAM is used for IPv6 multicast forwarding lookups. To increase TCAM capacity for handling IPv6 multicast routes, you must use the **hw-module slot number tcam carve rx_ipv6_mcast v6-mcast-percentage** command in privileged EXEC mode, where *v6-mcast-percentage* specifies the percentage of TCAM hardware used by IPv6 multicast prefix.

For example, you can change the IPv6 multicast region from 1 percent (default) to 16 percent of the TCAM hardware by reallocating the NetFlow region from 35 percent (default) to 20 percent as follows:

```
Router# hw-module slot 3 tcam carve rx_ipv6_mcast 16  
Router# hw-module slot 3 tcam carve rx_nf 20
```

IPv6 multicast hardware forwarding is support on the Cisco Catalyst 6500 and 7600 series in Cisco IOS Release 12.2(18)SXE.

Information About Implementing IPv6 Multicast

To configure IPv6 multicast, you need to understand the following concepts:

- [IPv6 Multicast Overview, page 4](#)
- [IPv6 Multicast Addressing, page 4](#)
- [IPv6 Multicast Routing Implementation, page 7](#)
- [Multicast Listener Discovery Protocol for IPv6, page 7](#)
- [Protocol Independent Multicast, page 9](#)
- [Static Mroutes, page 16](#)
- [MRIB, page 16](#)
- [MFIB, page 16](#)
- [IPv6 Multicast Process Switching and Fast Switching, page 17](#)
- [Multiprotocol BGP for the IPv6 Multicast Address Family, page 18](#)

IPv6 Multicast Overview

An IPv6 multicast group is an arbitrary group of receivers that want to receive a particular data stream. This group has no physical or geographical boundaries—receivers can be located anywhere on the Internet or in any private network. Receivers that are interested in receiving data flowing to a particular group must join the group by signaling their local router. This signaling is achieved with the MLD protocol.

Routers use the MLD protocol to learn whether members of a group are present on their directly attached subnets. Hosts join multicast groups by sending MLD report messages. The network then delivers data to a potentially unlimited number of receivers, using only one copy of the multicast data on each subnet. IPv6 hosts that wish to receive the traffic are known as group members.

Packets delivered to group members are identified by a single multicast group address. Multicast packets are delivered to a group using best-effort reliability, just like IPv6 unicast packets.

The multicast environment consists of senders and receivers. Any host, regardless of whether it is a member of a group, can send to a group. However, only the members of a group receive the message.

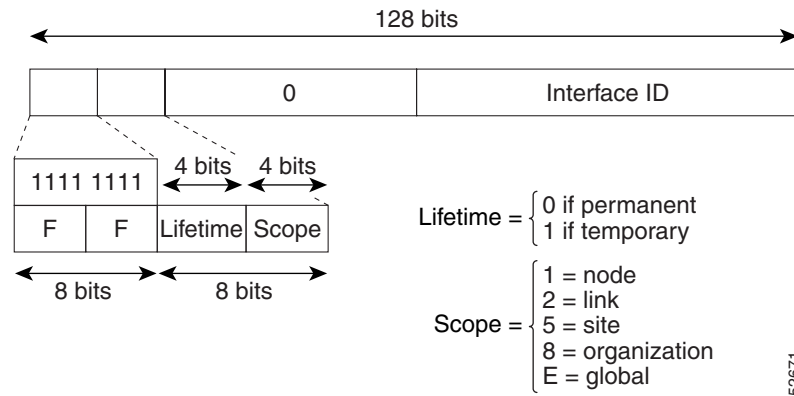
A multicast address is chosen for the receivers in a multicast group. Senders use that address as the destination address of a datagram to reach all members of the group.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time.

How active a multicast group is, its duration, and its membership can vary from group to group and from time to time. A group that has members may have no activity.

IPv6 Multicast Addressing

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that typically belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope. [Figure 1](#) shows the format of the IPv6 multicast address.

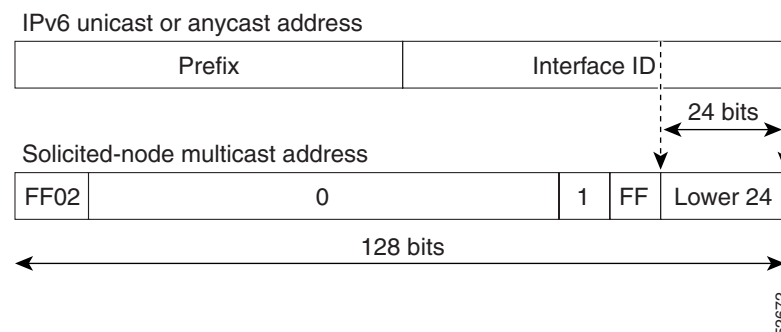
Figure 1 IPv6 Multicast Address Format

IPv6 nodes (hosts and routers) are required to join (receive packets destined for) the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:0:1 (scope is link-local)
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address (see [Figure 2](#)). For example, the solicited-node multicast address corresponding to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

Figure 2 IPv6 Solicited-Node Multicast Address Format**Note**

There are no broadcast addresses in IPv6. IPv6 multicast addresses are used instead of broadcast addresses.

IPv6 Multicast Groups

An IPv6 address must be configured on an interface for the interface to forward IPv6 traffic. Configuring a site-local or global IPv6 address on an interface automatically configures a link-local address and activates IPv6 for that interface. Additionally, the configured interface automatically joins the following required multicast groups for that link:

- Solicited-node multicast group FF02:0:0:0:1:FF00::/104 for each unicast and anycast address assigned to the interface
- All-nodes link-local multicast group FF02::1
- All-routers link-local multicast group FF02::2



Note The solicited-node multicast address is used in the neighbor discovery process.

For further information on configuring IPv6 addresses, refer to [Implementing IPv6 Addressing and Basic Connectivity](#).

Scoped Address Architecture

IPv6 includes support for global and nonglobal addresses. This section describes the usage of IPv6 addresses of different scopes.

A scope zone, or a simply a zone, is a connected region of topology of a given scope. For example, the set of links connected by routers within a particular site, and the interfaces attached to those links, comprise a single zone of site-local scope.

A zone is a particular instance of a topological region (for example, Zone1's site or Zone2's site), whereas a scope is the size of a topological region (for example, a site or a link). The zone to which a particular nonglobal address pertains is not encoded in the address itself, but rather is determined by context, such as the interface from which it is sent or received. Therefore, addresses of a given nonglobal scope may be reused in different zones of that scope. For example, Zone1's site and Zone2's site may each contain a node with site-local address FEC0::1.

Zones of the different scopes are instantiated as follows:

- Each link, and the interfaces attached to that link, comprises a single zone of link-local scope (for both unicast and multicast).
- There is a single zone of global scope (for both unicast and multicast), comprising all the links and interfaces in the Internet.
- The boundaries of zones of scope other than interface-local, link-local, and global must be defined and configured by network administrators. A site boundary serves as such for both unicast and multicast.

Zone boundaries are relatively static features and do not change in response to short-term changes in topology. Therefore, the requirement that the topology within a zone be "connected" is intended to include links and interfaces that may be connected only occasionally. For example, a residential node or network that obtains Internet access by dialup to an employer's site may be treated as part of the employer's site-local zone even when the dialup link is disconnected. Similarly, a failure of a router, interface, or link that causes a zone to become partitioned does not split that zone into multiple zones; rather, the different partitions are still considered to belong to the same zone.

Zones have the following additional properties:

- Zone boundaries cut through nodes, not links (the global zone has no boundary, and the boundary of an interface-local zone encloses just a single interface.)
- Zones of the same scope cannot overlap; that is, they can have no links or interfaces in common.
- A zone of a given scope (less than global) falls completely within zones of larger scope; that is, a smaller scope zone cannot include more topology than any larger scope zone with which it shares any links or interfaces.
- Each interface belongs to exactly one zone of each possible scope.

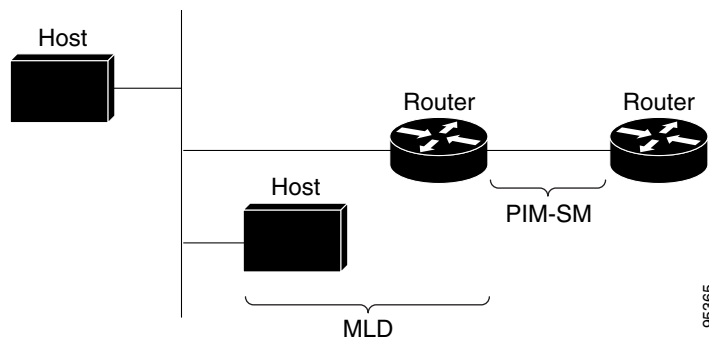
IPv6 Multicast Routing Implementation

The Cisco IOS software supports the following protocols to implement IPv6 multicast routing:

- MLD for IPv6. MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD: MLD version 1 is based on version 2 of the Internet Group Management Protocol (IGMP) for IPv4, and MLD version 2 is based on version 3 of the IGMP for IPv4. IPv6 multicast for Cisco IOS software uses both MLD version 2 and MLD version 1. MLD version 2 is fully backward-compatible with MLD version 1 (described in RFC 2710). Hosts that support only MLD version 1 will interoperate with a router running MLD version 2. Mixed LANs with both MLD version 1 and MLD version 2 hosts are likewise supported.
- PIM-SM is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs.
- PIM in Source Specific Multicast (PIM-SSM) is similar to PIM-SM with the additional ability to report interest in receiving packets from specific source addresses (or from all but the specific source addresses) to an IP multicast address.

Figure 3 shows where MLD and PIM-SM operate within the IPv6 multicast environment.

Figure 3 IPv6 Multicast Routing Protocols Supported for IPv6



Multicast Listener Discovery Protocol for IPv6

To start implementing multicasting in the campus network, users must first define who receives the multicast. The MLD protocol is used by IPv6 routers to discover the presence of multicast listeners (for example, nodes that want to receive multicast packets) on their directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes. It is used for

discovering local group and source-specific group membership. The MLD protocol provides a means to automatically control and limit the flow of multicast traffic throughout your network with the use of special multicast queriers and hosts.

The difference between multicast queriers and hosts is as follows:

- A querier is a network device, such as a router, that sends query messages to discover which network devices are members of a given multicast group.
- A host is a receiver, including routers, that send report messages to inform the querier of a host membership.

A set of queriers and hosts that receive multicast data streams from the same source is called a multicast group. Queriers and hosts use MLD reports to join and leave multicast groups and to begin receiving group traffic.

MLD uses the Internet Control Message Protocol (ICMP) to carry its messages. All MLD messages are link-local with a hop limit of 1, and they all have the router alert option set. The router alert option implies an implementation of the hop-by-hop option header.

MLD has three types of messages:

- Query—General, group-specific, and multicast-address-specific. In a query message, the multicast address field is set to 0 when MLD sends a general query. The general query learns which multicast addresses have listeners on an attached link.

Group-specific and multicast-address-specific queries are the same. A group address is a multicast address.

- Report—In a report message, the multicast address field is that of the specific IPv6 multicast address to which the sender is listening.
- Done—In a done message, the multicast address field is that of the specific IPv6 multicast address to which the source of the MLD message is no longer listening.

An MLD report must be sent with a valid IPv6 link-local source address, or the unspecified address (::), if the sending interface has not yet acquired a valid link-local address. Sending reports with the unspecified address is allowed to support the use of IPv6 multicast in the Neighbor Discovery Protocol.

For stateless autoconfiguration, a node is required to join several IPv6 multicast groups in order to perform duplicate address detection (DAD). Prior to DAD, the only address the reporting node has for the sending interface is a tentative one, which cannot be used for communication. Therefore, the unspecified address must be used.

MLD states that result from MLD version 2 or MLD version 1 membership reports can be limited globally or by interface. The MLD group limits feature provides protection against denial of service (DoS) attacks caused by MLD packets. Membership reports in excess of the configured limits will not be entered in the MLD cache, and traffic for those excess membership reports will not be forwarded.

MLD provides support for source filtering. Source filtering allows a node to report interest in listening to packets only from specific source addresses (as required to support SSM), or from all addresses except specific source addresses sent to a particular multicast address.

When a host using MLD version 1 sends a leave message, the router needs to send query messages to reconfirm that this host was the last MLD version 1 host joined to the group before it can stop forwarding traffic. This function takes about 2 seconds. This “leave latency” is also present in IGMP version 2 for IPv4 multicast.

MLD Access Group

The MLD access group provides receiver access control in Cisco IOS IPv6 multicast routers. This feature limits the list of groups a receiver can join, and it allows or denies sources used to join SSM channels.

Explicit Tracking of Receivers

The explicit tracking feature allows a router to track the behavior of the hosts within its IPv6 network. This feature also enables the fast leave mechanism to be used with MLD version 2 host reports.

Multicast User Authentication and Profile Support

IPv6 multicast by design allows any host in the network to become a receiver or a source for a multicast group. Therefore, multicast access control is needed to control multicast traffic in the network. Access control functionality consists mainly of source access control and accounting, receiver access control and accounting, and provisioning of this access control mechanism.

Multicast access control provides an interface between multicast and authentication, authorization, and accounting (AAA) for provisioning, authorizing, and accounting at the last-hop router, receiver access control functions in multicast, and group or channel disabling capability in multicast.

When you deploy a new multicast service environment, it is necessary to add user authentication and provide a user profile download on a per-interface basis. The use of AAA and IPv6 multicast supports user authentication and downloading of the user profile in a multicast environment.

The event that triggers the download of a multicast cast access-control profile from the RADIUS server to the access router is arrival of an MLD join on the access router. When this event occurs, a user can case the authorization cache to time out and request download periodically or use an appropriate multicast clear command to trigger a new download in case of profile changes.

Accounting occurs via RADIUS accounting. Start and stop accounting records are sent to the RADIUS server from the access router. In order for you to track resource consumption on a per-stream basis, these accounting records provide information about the multicast source and group. The start record is sent when the last-hop router receives a new MLD report, and the stop record is sent upon MLD leave or if the group or channel is deleted for any reason.

Protocol Independent Multicast

Protocol Independent Multicast (PIM) is used between routers so that they can track which multicast packets to forward to each other and to their directly connected LANs. PIM works independently of the unicast routing protocol to perform send or receive multicast route updates like other protocols.

Regardless of which unicast routing protocols are being used in the LAN to populate the unicast routing table, Cisco IOS PIM uses the existing unicast table content to perform the Reverse Path Forwarding (RPF) check instead of building and maintaining its own separate routing table.

You can configure IPv6 multicast to use either PIM-SM or PIM-SSM operation, or you can use both PIM-SM and PIM-SSM together in your network.

PIM-Sparse Mode

IPv6 multicast provides support for intradomain multicast routing using PIM-SM. PIM-SM uses unicast routing to provide reverse-path information for multicast tree building, but it is not dependent on any particular unicast routing protocol.

PIM-SM is used in a multicast network when relatively few routers are involved in each multicast and these routers do not forward multicast packets for a group, unless there is an explicit request for the traffic. PIM-SM distributes information about active sources by forwarding data packets on the shared tree. PIM-SM initially uses shared trees, which requires the use of an RP.

Requests are accomplished via PIM joins, which are sent hop by hop toward the root node of the tree. The root node of a tree in PIM-SM is the RP in the case of a shared tree or the first-hop router that is directly connected to the multicast source in the case of a shortest path tree (SPT). The RP keeps track of multicast groups and the hosts that send multicast packets are registered with the RP by that host's first-hop router.

As a PIM join travels up the tree, routers along the path set up multicast forwarding state so that the requested multicast traffic will be forwarded back down the tree. When multicast traffic is no longer needed, a router sends a PIM prune up the tree toward the root node to prune (or remove) the unnecessary traffic. As this PIM prune travels hop by hop up the tree, each router updates its forwarding state appropriately. Ultimately, the forwarding state associated with a multicast group or source is removed.

A multicast data sender sends data destined for a multicast group. The designated router (DR) of the sender takes those data packets, unicast-encapsulates them, and sends them directly to the RP. The RP receives these encapsulated data packets, de-encapsulates them, and forwards them onto the shared tree. The packets then follow the (*, G) multicast tree state in the routers on the RP tree, being replicated wherever the RP tree branches, and eventually reaching all the receivers for that multicast group. The process of encapsulating data packets to the RP is called registering, and the encapsulation packets are called PIM register packets.

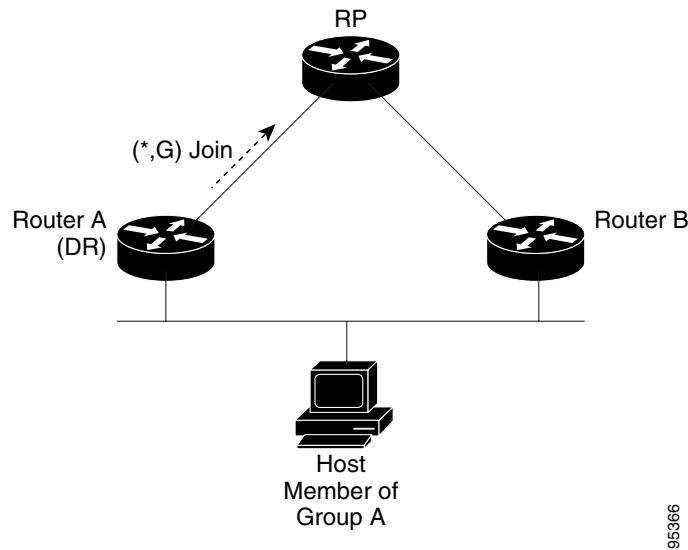
Designated Router

Cisco routers use PIM-SM to forward multicast traffic and follow an election process to select a designated router when there is more than one router on a LAN segment.

The designated router is responsible for sending PIM register and PIM join and prune messages toward the RP to inform it about host group membership.

If there are multiple PIM-SM routers on a LAN, a designated router must be elected to avoid duplicating multicast traffic for connected hosts. The PIM router with the highest IPv6 address becomes the DR for the LAN unless you choose to force the DR election by use of the **ipv6 pim dr-priority** command. This command allows you to specify the DR priority of each router on the LAN segment (default priority = 1) so that the router with the highest priority will be elected as the DR. If all routers on the LAN segment have the same priority, then the highest IPv6 address is again used as the tiebreaker.

Figure 4 illustrates what happens on a multiaccess segment. Router A and Router B are connected to a common multiaccess Ethernet segment with Host A as an active receiver for Group A. Only Router A, operating as the DR, sends joins to the RP to construct the shared tree for Group A. If Router B was also permitted to send (*, G) joins to the RP, parallel paths would be created and Host A would receive duplicate multicast traffic. Once Host A begins to source multicast traffic to the group, the DR's responsibility is to send register messages to the RP. If both routers were assigned the responsibility, the RP would receive duplicate multicast packets.

Figure 4 Designated Router Election on a Multiaccess Segment

If the DR should fail, the PIM-SM provides a way to detect the failure of Router A and elect a failover DR. If the DR (Router A) became inoperable, Router B would detect this situation when its neighbor adjacency with Router A timed out. Because Router B has been hearing MLD membership reports from Host A, it already has MLD state for Group A on this interface and would immediately send a join to the RP when it became the new DR. This step reestablishes traffic flow down a new branch of the shared tree via Router B. Additionally, if Host A were sourcing traffic, Router B would initiate a new register process immediately after receiving the next multicast packet from Host A. This action would trigger the RP to join the SPT to Host A via a new branch through Router B.

**Tip**

Two PIM routers are neighbors if there is a direct connection between them. To display your PIM neighbors, use the **show ipv6 pim neighbor** command in privileged EXEC mode.

**Note**

DR election process is required only on multiaccess LANs. The last-hop router directly connected to the host is the DR.

Rendezvous Point

IPv6 PIM provides embedded RP support. Embedded RP support allows the router to learn RP information using the multicast group destination address instead of the statically configured RP. For routers that are the RP, the router must be statically configured as the RP.

The router searches for embedded RP group addresses in MLD reports or PIM messages and data packets. On finding such an address, the router learns the RP for the group from the address itself. It then uses this learned RP for all protocol activity for the group. For routers that are the RP, the router is advertised as an embedded RP must be configured as the RP.

To select a static RP over an embedded RP, the specific embedded RP group range or mask must be configured in the access list of the static RP. When PIM is configured in sparse mode, you must also choose one or more routers to operate as an RP. An RP is a single common root placed at a chosen point of a shared distribution tree and is configured statically in each box.

PIM DRs forward data from directly connected multicast sources to the RP for distribution down the shared tree. Data is forwarded to the RP in one of two ways:

- Data is encapsulated in register packets and unicast directly to the RP by the first-hop router operating as the DR.
- If the RP has itself joined the source tree, it is multicast-forwarded per the RPF forwarding algorithm described in the “[PIM-Sparse Mode](#)” section.

The RP address is used by first-hop routers to send PIM register messages on behalf of a host sending a packet to the group. The RP address is also used by last-hop routers to send PIM join and prune messages to the RP to inform it about group membership. You must configure the RP address on all routers (including the RP router).

A PIM router can be an RP for more than one group. Only one RP address can be used at a time within a PIM domain for a certain group. The conditions specified by the access list determine for which groups the router is an RP.

IPv6 multicast supports the PIM accept register feature, which is the ability to perform PIM-SM register message filtering at the RP. The user can match an access list or compare the AS path for the registered source with the AS path specified in a route map.

IPv6 BSR

PIM routers in a domain must be able to map each multicast group to the correct RP address. The BSR protocol for PIM-SM provides a dynamic, adaptive mechanism to distribute group-to-RP mapping information rapidly throughout a domain. With the IPv6 BSR feature, if an RP becomes unreachable, it will be detected and the mapping tables will be modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.

Every PIM-SM multicast group needs to be associated with the IP or IPv6 address of an RP. When a new multicast sender starts sending, its local DR will encapsulate these data packets in a PIM register message and send them to the RP for that multicast group. When a new multicast receiver joins, its local DR will send a PIM join message to the RP for that multicast group. When any PIM router sends a (*, G) join message, the PIM router needs to know which is the next router toward the RP so that G (Group) can send a message to that router. Also, when a PIM router is forwarding data packets using (*, G) state, the PIM router needs to know which is the correct incoming interface for packets destined for G, because it needs to reject any packets that arrive on other interfaces.

A small set of routers from a domain are configured as candidate bootstrap routers (C-BSRs) and a single BSR is selected for that domain. A set of routers within a domain are also configured as candidate RPs (C-RPs); typically, these routers are the same routers that are configured as C-BSRs. Candidate RPs periodically unicast candidate-RP-advertisement (C-RP-Adv) messages to the BSR of that domain, advertising their willingness to be an RP. A C-RP-Adv message includes the address of the advertising C-RP, and an optional list of group addresses and mask length fields, indicating the group prefixes for which the candidacy is advertised. The BSR then includes a set of these C-RPs, along with their corresponding group prefixes, in bootstrap messages (BSMs) it periodically originates. BSMs are distributed hop-by-hop throughout the domain.

The IPv6 BSR ability to configure RP mapping allows IPv6 multicast routers to be statically configured to announce scope-to-RP mappings directly from the BSR instead of learning them from candidate-RP messages. Announcing RP mappings from the BSR is useful in several situations:

- When an RP address never changes because there is only a single RP or the group range uses an anycast RP, it may be less complex to configure the RP address announcement statically on the candidate BSRs.

- When an RP address is a virtual RP address (such as when using bidirectional PIM), it cannot be learned by the BSR from a candidate-RP. Instead, the virtual RP address must be configured as an announced RP on the candidate BSRs.

Cisco IOS IPv6 routers provide support for the RPF flooding of BSR packets so that a Cisco IOS IPv6 router will not disrupt the flow of BSMs. The router will recognize and parse enough of the BSM to identify the BSR address. The router performs an RPF check for this BSR address and forwards the packet only if it is received on the RPF interface. The router also creates a BSR entry containing RPF information to use for future BSMs from the same BSR. When BSMs from a given BSR are no longer received, the BSR entry is timed out.

Bidirectional BSR support allows bidirectional RPs to be advertised in C-RP messages and bidirectional ranges in the BSM. All routers in a system must be able to use the bidirectional range in the BSM; otherwise, the bidirectional RP feature will not function.

BSR provides scoped zone support by distributing group-to-RP mappings in networks using administratively scoped multicast. The user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the user's domain.

For BSR to function correctly with administrative scoping, a BSR and at least one C-RP must be within every administratively scoped region. Administratively scoped zone boundaries must be configured at the zone border routers (ZBRs), because they need to filter PIM join messages that might inadvertently cross the border due to error conditions. In addition, at least one C-BSR within the administratively scoped zone must be configured to be a C-BSR for the administratively scoped zone's address range.

A separate BSR election will then take place (using BSMs) for every administratively scoped range, plus one for the global range. Administratively scoped ranges are identified in the BSM because the group range is marked to indicate that this is an administrative scope range, not just a range that a particular set of RPs is configured to handle.

Unless the C-RP is configured with a scope, it discovers the existence of the administratively scoped zone and its group range through reception of a BSM from the scope zone's elected BSR containing the scope zone's group range. A C-RP stores each elected BSR's address and the administratively scoped range contained in its BSM. It separately unicasts C-RP-Adv messages to the appropriate BSR for every administratively scoped range within which it is willing to serve as an RP.

All PIM routers within a PIM bootstrap domain where administratively scoped ranges are in use must be able to receive BSMs and store the winning BSR and RP set for all administratively scoped zones that apply.

PIM-Source Specific Multicast

PIM-SSM is the routing protocol that supports the implementation of SSM and is derived from PIM-SM. However, unlike PIM-SM where data from all multicast sources are sent when there is a PIM join, the SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, thus optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. Further, instead of the use of RP and shared trees, SSM uses information found on source addresses for a multicast group. This information is provided by receivers through the source addresses relayed to the last-hop routers by MLD membership reports, resulting in shortest-path trees directly to the sources.

In SSM, delivery of datagrams is based on (S, G) channels. Traffic for one (S, G) channel consists of datagrams with an IPv6 unicast source address S and the multicast group address G as the IPv6 destination address. Systems will receive this traffic by becoming members of the (S, G) channel. Signaling is not required, but receivers must subscribe or unsubscribe to (S, G) channels to receive or not receive traffic from specific sources.

MLD version 2 is required for SSM to operate. MLD allows the host to provide source information. Before SSM will run with MLD, SSM must be supported in the Cisco IOS IPv6 router, the host where the application is running, and the application itself.

SSM Mapping for IPv6

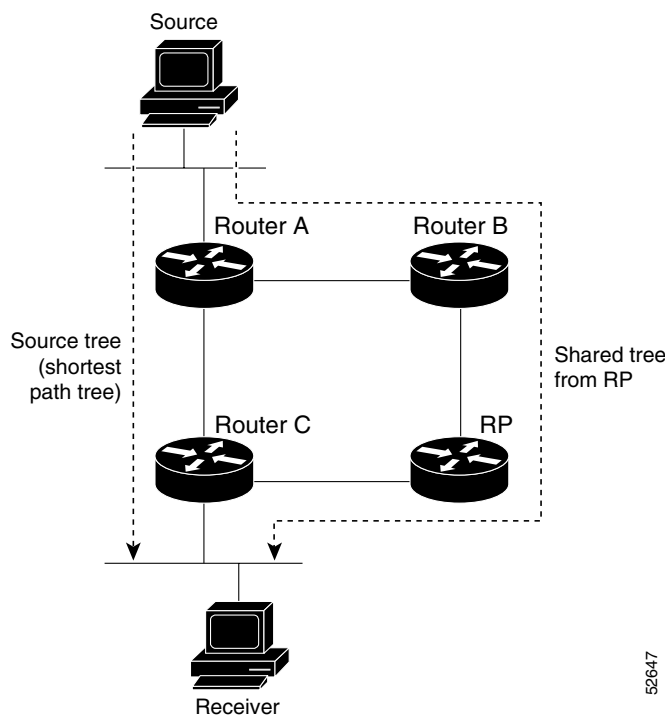
SSM mapping for IPv6 supports both static and dynamic Domain Name System (DNS) mapping for MLD version 1 receivers. This feature allows deployment of IPv6 SSM with hosts that are incapable of providing MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application.

SSM mapping allows the router to look up the source of a multicast MLD version 1 report either in the running configuration of the router or from a DNS server. The router can then initiate an (S, G) join toward the source.

PIM Shared Tree and Source Tree (Shortest-Path Tree)

By default, members of a group receive data from senders to the group across a single data distribution tree rooted at the RP. This type of distribution tree is called shared tree or rendezvous point tree (RPT), as illustrated in [Figure 5](#). Data from senders is delivered to the RP for distribution to group members joined to the shared tree.

Figure 5 Shared Tree and Source Tree (Shortest Path Tree)



If the data threshold warrants, leaf routers on the shared tree may initiate a switch to the data distribution tree rooted at the source. This type of distribution tree is called a shortest path tree or source tree. By default, the Cisco IOS software switches to a source tree upon receiving the first data packet from a source.

The following process describes the move from shared tree to source tree in more detail:

1. Receiver joins a group; leaf Router C sends a join message toward the RP.

2. RP puts the link to Router C in its outgoing interface list.
3. Source sends the data; Router A encapsulates the data in the register and sends it to the RP.
4. RP forwards the data down the shared tree to Router C and sends a join message toward the source. At this point, data may arrive twice at Router C, once encapsulated and once natively.
5. When data arrives natively (unencapsulated) at the RP, the RP sends a register-stop message to Router A.
6. By default, receipt of the first data packet prompts Router C to send a join message toward the source.
7. When Router C receives data on (S, G), it sends a prune message for the source up the shared tree.
8. RP deletes the link to Router C from the outgoing interface of (S, G).
9. RP triggers a prune message toward the source.

Join and prune messages are sent for sources and RPs. They are sent hop-by-hop and are processed by each PIM router along the path to the source or RP. Register and register-stop messages are not sent hop-by-hop. They are sent by the designated router that is directly connected to a source and are received by the RP for the group.

Reverse Path Forwarding

Reverse-path forwarding is used for forwarding multicast datagrams. It functions as follows:

- If a router receives a datagram on an interface it uses to send unicast packets to the source, the packet has arrived on the RPF interface.
- If the packet arrives on the RPF interface, a router forwards the packet out the interfaces present in the outgoing interface list of a multicast routing table entry.
- If the packet does not arrive on the RPF interface, the packet is silently discarded to prevent loops.

PIM uses both source trees and RP-rooted shared trees to forward datagrams; the RPF check is performed differently for each, as follows:

- If a PIM router has source-tree state (that is, an (S, G) entry is present in the multicast routing table), the router performs the RPF check against the IPv6 address of the source of the multicast packet.
- If a PIM router has shared-tree state (and no explicit source-tree state), it performs the RPF check on the RP's address (which is known when members join the group).

Sparse-mode PIM uses the RPF lookup function to determine where it needs to send joins and prunes. (S, G) joins (which are source-tree states) are sent toward the source. (*, G) joins (which are shared-tree states) are sent toward the RP.

Routable Address Hello Option

When an IPv6 interior gateway protocol is used to build the unicast routing table, the procedure to detect the upstream router address assumes the address of a PIM neighbor is always same as the address of the next-hop router, as long as they refer to the same router. However, it may not be the case when a router has multiple addresses on a link.

Two typical situations can lead to this situation for IPv6. The first situation can occur when the unicast routing table is not built by an IPv6 interior gateway protocol such as multicast BGP. The second situation occurs when the address of an RP shares a subnet prefix with downstream routers (note that the RP router address has to be domain-wide and therefore cannot be a link-local address).

The routable address hello option allows the PIM protocol avoid such situations by adding a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised. When a PIM router finds an upstream router for some address, the result of RPF calculation is compared with the addresses in this option, in addition to PIM neighbor's address itself. Because this option includes all the possible addresses of a PIM router on that link, it always includes the RPF calculation result if it refers to the PIM router supporting this option.

Bidirectional PIM

Bidirectional PIM allows multicast routers to keep reduced state information, as compared with unidirectional shared trees in PIM-SM. Bidirectional shared trees convey data from sources to the RP and distribute them from the RP to the receivers. Unlike PIM-SM, bidirectional PIM does not switch over to the source tree, and there is no register encapsulation of data from the source to the RP.

Bidirectional PIM offers advantages when there are many moderate or low-rate sources. However, the bidirectional source trees have worse delay characteristics than do the source trees built in PIM-SM.

Only static configuration of bidirectional RPs is supported in IPv6.

Static Mroutes

IPv6 static mroutes behave much in the same way as do IPv6 static routes. IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support. Static mroutes support equal-cost multipath mroutes, and they also support unicast-only static routes.

For further information on IPv6 static routes, see [Implementing Static Routes for IPv6](#).

MRIB

The Multicast Routing Information Base (MRIB) is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). Its main function is to provide independence between routing protocols and the Multicast Forwarding Information Base (MFIB). It also acts as a coordination and communication point among its clients.

Routing clients use the services provided by the MRIB to instantiate routing entries and retrieve changes made to routing entries by other clients. Besides routing clients, MRIB also has forwarding clients (MFIB instances) and special clients such as MLD. MFIB retrieves its forwarding entries from MRIB and notifies the MRIB of any events related to packet reception. These notifications can either be explicitly requested by routing clients or spontaneously generated by the MFIB.

Another important function of the MRIB is to allow for the coordination of multiple routing clients in establishing multicast connectivity within the same multicast session. MRIB also allows for the coordination between MLD and routing protocols.

MFIB

The MFIB is a platform-independent and routing-protocol-independent library for IPv6 software. Its main purpose is to provide a Cisco IOS platform with an interface with which to read the IPv6 multicast forwarding table and notifications when the forwarding table changes. The information provided by the MFIB has clearly defined forwarding semantics and is designed to make it easy for the platform to translate to its specific hardware or software forwarding mechanisms.

When routing or topology changes occur in the network, the IPv6 routing table is updated, and those changes are reflected in the MFIB. The MFIB maintains next-hop address information based on the information in the IPv6 routing table. Because there is a one-to-one correlation between MFIB entries and routing table entries, the MFIB contains all known routes and eliminates the need for route cache maintenance that is associated with switching paths such as fast switching and optimum switching.

Distributed MFIB

Distributed MFIB (dMFIB) is used to switch multicast IPv6 packets on distributed platforms. dMFIB may also contain platform-specific information on replication across line cards. The basic MFIB routines that implement the core of the forwarding logic are common to all forwarding environments.

dMFIB implements the following functions:

- Distributes a copy of the MFIB to the line cards.
- Relays data-driven protocol events generated in the line cards to PIM.
- Provides an MFIB platform application program interface (API) to propagate MFIB changes to platform-specific code responsible for programming the hardware acceleration engine. This API also includes entry points to switch a packet in software (necessary if the packet is triggering a data-driven event) and to upload traffic statistics to the software.
- Provides hooks to allow clients residing on the RP to read traffic statistics on demand. (dMFIB does not periodically upload these statistics to the RP.)

The combination of dMFIB and MRIB subsystems also allows the router to have a “customized” copy of the MFIB database in each line card and to transport MFIB-related platform-specific information from the RP to the line cards.

IPv6 Multicast Process Switching and Fast Switching

A unified MFIB is used to provide both fast switching and process switching support for PIM-SM and PIM-SSM in IPv6 multicast. In process switching, the Route Processor must examine, rewrite, and forward each packet. The packet is first received and copied into the system memory. The router then looks up the Layer 3 network address in the routing table. The Layer 2 frame is then rewritten with the next-hop destination address and sent to the outgoing interface. The RP also computes the cyclic redundancy check (CRC). This switching method is the least scalable method for switching IPv6 packets.

IPv6 multicast fast switching allows routers to provide better packet forwarding performance than process switching. Information conventionally stored in a route cache is stored in several data structures for IPv6 multicast switching. The data structures provide optimized lookup for efficient packet forwarding.

In IPv6 multicast forwarding, the first packet is fast-switched if the PIM protocol logic allows it. Also in IPv6 multicast fast switching, the MAC encapsulation header is precomputed. IPv6 multicast fast switching uses the MFIB to make IPv6 destination prefix-based switching decisions. In addition to the MFIB, IPv6 multicast fast switching uses adjacency tables to prepend Layer 2 addressing information. The adjacency table maintains Layer 2 next-hop addresses for all MFIB entries.

The adjacency table is populated as adjacencies are discovered. Each time an adjacency entry is created (such as through ARP), a link-layer header for that adjacent node is precomputed and stored in the adjacency table. Once a route is determined, it points to a next hop and corresponding adjacency entry. It is subsequently used for encapsulation during switching of packets.

A route might have several paths to a destination prefix, such as when a router is configured for simultaneous load balancing and redundancy. For each resolved path, a pointer is added for the adjacency corresponding to the next-hop interface for that path. This mechanism is used for load balancing across several paths.

Multiprotocol BGP for the IPv6 Multicast Address Family

The multicast BGP for the IPv6 multicast address family feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. IPv6 enhancements to multicast BGP include support for an IPv6 multicast address family and network layer reachability information (NLRI) and next hop (the next router in the path to the destination) attributes that use IPv6 addresses.

Multicast BGP is an enhanced BGP that allows the deployment of interdomain IPv6 multicast. Multicast BGP carries routing information for multiple network layer protocol address families; for example, IPv6 address family and for IPv6 multicast routes. The IPv6 multicast address family contains routes used for RPF lookup by the IPv6 PIM protocol, and multicast BGP IPv6 provides for interdomain transport of the same. Users must use multicast BGP for IPv6 multicast when using IPv6 multicast with BGP because the unicast multicast BGP learned routes will not be used for IPv6 multicast.

Multicast BGP functionality provided through a separate address family context. A subsequent address family identifier (SAFI) provides information about the type of the network layer reachability information that is carried in the attribute. Multiprotocol BGP unicast uses SAFI 1 messages, and multiprotocol BGP multicast uses SAFI 2 messages.

A separate BGP routing table is maintained to configure incongruent policies and topologies (for example, IPv6 unicast and multicast) by using IPv6 multicast RPF lookup. Multicast RPF lookup is very similar to the IP unicast route lookup.

No MRIB is associated with the IPv6 multicast BGP table. However, IPv6 multicast BGP operates on the unicast IPv6 RIB when needed. Multicast BGP does not insert or update routes into the IPv6 unicast RIB.

How to Implement IPv6 Multicast

The tasks in the following sections explain how to implement IPv6 multicast:

- [Enabling IPv6 Multicast Routing, page 19](#)
- [Configuring the MLD Protocol, page 19](#)
- [Configuring PIM, page 29](#)
- [Configuring a BSR, page 37](#)
- [Configuring SSM Mapping, page 41](#)
- [Configuring Static Mroutes, page 42](#)
- [Configuring IPv6 Multiprotocol BGP, page 44](#)
- [Using MFIB in IPv6 Multicast, page 54](#)
- [Disabling Default Features in IPv6 Multicast, page 56](#)

Enabling IPv6 Multicast Routing

This task explains how to enable IPv6 multicast routing on all interfaces and to enable multicast forwarding for PIM and MLD on all enabled interfaces of the router.

Prerequisites

In order to enable IPv6 multicast routing on a router, you must first enable IPv6 unicast routing on the router. For information on how to enable IPv6 unicast routing on a router, refer to [Implementing IPv6 Addressing and Basic Connectivity](#).

If you are already using an IPv6 unicast router, use the following tasks to enable IPv6 multicast routing and configure IPv6 multicast routing options.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 multicast-routing Example: Router(config)# ipv6 multicast-routing	Enables multicast routing on all IPv6-enabled interfaces and enables multicast forwarding for PIM and MLD on all enabled interfaces of the router.

Configuring the MLD Protocol

The following tasks describe how to customize and verify the MLD protocol.

- [Customizing and Verifying MLD on an Interface, page 20](#)
- [Implementing MLD Group Limits, page 22](#)
- [Configuring Explicit Tracking of Receivers to Track Host Behavior, page 24](#)
- [Configuring Multicast User Authentication and Profile Support, page 25](#)
- [Resetting the MLD Traffic Counters, page 28](#)
- [Clearing the MLD Interface Counters, page 29](#)

Customizing and Verifying MLD on an Interface


Use the following tasks to customize MLD and verify MLD information on each interface, if desired.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld join-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]}
5. **ipv6 mld access-group** *access-list-name*
6. **ipv6 mld static-group** [*group-address*] [**include** | **exclude**] {*source-address* | **source-list** [*acl*]}
7. **ipv6 mld query-max-response-time** *seconds*
8. **ipv6 mld query-timeout** *seconds*
9. **ipv6 mld query-interval** *seconds*
10. **exit**
11. **show ipv6 mld groups** [**link-local**] [*group-name* | *group-address*] [*interface-type* | *interface-number*] [**detail** | **explicit**]
12. **show ipv6 mld groups summary**
13. **show ipv6 mld interface** [*type number*]
14. **debug ipv6 mld** [*group-name* | *group-address* | *interface-type*]
15. **debug ipv6 mld explicit** [*group-name* | *group-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	interface <i>type number</i>	Specifies an interface type and number, and places the router in interface configuration mode.
	Example: Router(config)# interface FastEthernet 1/0	

	Command or Action	Purpose
Step 4	ipv6 mld join-group [group-address] [include exclude] {source-address source-list [acl]} Example: Router(config-if)# ipv6 mld join-group FF04::12 exclude 2001:0DB8::10::11	Configures MLD reporting for a specified group and source.
Step 5	ipv6 mld access-group access-list-name Example: Router(config-if)# ipv6 access-list acc-grp-1	Allows the user to perform IPv6 multicast receiver access control.
Step 6	ipv6 mld static-group [group-address] [include exclude] {source-address source-list [acl]} Example: Router(config-if)# ipv6 mld static-group ff04::10 include 100::1	Statically forwards traffic for the multicast group onto a specified interface and cause the interface to behave as if a MLD joiner were present on the interface.
Step 7	ipv6 mld query-max-response-time seconds Example: Router(config-if)# ipv6 mld query-max-response-time 20	Configures the maximum response time advertised in MLD queries.
Step 8	ipv6 mld query-timeout seconds Example: Router(config-if)# ipv6 mld query-timeout 130	Configures the timeout value before the router takes over as the querier for the interface.
Step 9	ipv6 mld query-interval seconds Example: Router(config-if)# ipv6 mld query-interval 60	Configures the frequency at which the Cisco IOS software sends MLD host-query messages. <div>  Caution Changing this value may severely impact multicast forwarding. </div>
Step 10	exit Example: Router(config-if)# exit	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 11	show ipv6 mld groups [link-local] [group-name group-address] [interface-type interface-number] [detail explicit] Example: Router# show ipv6 mld groups FastEthernet 2/1	Displays the multicast groups that are directly connected to the router and that were learned through MLD.
Step 12	show ipv6 mld groups summary Example: Router# show ipv6 mld groups summary	Displays the number of (*, G) and (S, G) membership reports present in the MLD cache.

	Command or Action	Purpose
Step 13	show ipv6 mld interface [<i>type number</i>] Example: Router# show ipv6 mld interface FastEthernet 2/1	Displays multicast-related information about an interface.
Step 14	debug ipv6 mld [<i>group-name</i> <i>group-address</i> <i>interface-type</i>] Example: Router# debug ipv6 mld	Enables debugging on MLD protocol activity.
Step 15	debug ipv6 mld explicit [<i>group-name</i> <i>group-address</i>] Example: Router# debug ipv6 mld explicit	Displays information related to the explicit tracking of hosts.

Implementing MLD Group Limits

Per-interface and global MLD limits operate independently of each other. Both per-interface and global MLD limits can be configured on the same router. The number of MLD limits, globally or per interface, is not configured by default; the limits must be configured by the user. A membership report that exceeds either the per-interface or the global state limit is ignored.

The following tasks describe how to limit MLD states that result from MLD version 2 or MLD version 1 membership reports globally or per interface:

- [Implementing MLD Group Limits Globally, page 22](#)
- [Clearing the MLD Interface Counters, page 29](#)

Implementing MLD Group Limits Globally

This following example describes how to limit MLD groups globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld state-limit** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mld state-limit <i>number</i> Example: Router(config)# ipv6 mld state-limit 300	Limits the number of MLD states globally.

Implementing MLD Group Limits per Interface

This task describes how to limit MLD groups on a per-interface basis.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld limit** *number* [**except** *access-list*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 mld limit <i>number</i> [except <i>access-list</i>] Example: Router(config-if)# ipv6 mld limit 100	Limits the number of MLD states on a per-interface basis.

Configuring Explicit Tracking of Receivers to Track Host Behavior

This task enables the explicit tracking of receivers feature. The explicit tracking feature allows a router to track the behavior of the hosts within its IPv6 network and enables the fast leave mechanism to be used with MLD version 2 host reports.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 mld explicit-tracking** *access-list-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 mld explicit-tracking <i>access-list-name</i> Example: Router(config-if)# ipv6 mld explicit-tracking list1	Enables explicit tracking of hosts.

Configuring Multicast User Authentication and Profile Support

This section describes several tasks used to enable and configure the multicast user authentication and profile support feature.

Prerequisites

Before you configure multicast user authentication and profile support, you may configure the following receiver access control functions in IPv6 multicast.

- To limit MLD groups globally, see the [“Implementing MLD Group Limits Globally” section on page 22](#).
- To limit MLD groups on a per-interface basis, see the [“Implementing MLD Group Limits per Interface” section on page 23](#)
- To specify the MLD groups and sources allowed on an interface, see the [“Customizing and Verifying MLD on an Interface” section on page 20](#), step 5.

Restrictions

Before you configure multicast user authentication and profile support, you should be aware of the following restrictions:

- The port, interface, VC, or VLAN ID is the user or subscriber identity. User identity by hostname, user ID, or password is not supported.

To configure multicast user authentication and profile support, perform the following tasks:

- [Enabling AAA Access Control for IPv6 Multicast, page 25](#)
- [Specifying Method Lists and Enabling Multicast Accounting, page 26](#)
- [Disabling the Router from Receiving Unauthenticated Multicast Traffic, page 27](#)
- [Resetting Authorization Status on an MLD Interface, page 28](#)

Enabling AAA Access Control for IPv6 Multicast

The following example describes how to enable AAA access control.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	aaa new-model Example: Router(config)# aaa new-model	Enables the AAA access control system.

Specifying Method Lists and Enabling Multicast Accounting

The following task describes how to specify the method lists used for AAA authorization and accounting and how to enable multicast accounting on specified groups or channels on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa authorization multicast default** [*method3* | *method4*]
4. **aaa accounting multicast default** [*start-stop* | *stop-only*] [*broadcast*] [*method1*] [*method2*] [*method3*] [*method4*]
5. **interface** *type number*
6. **ipv6 multicast aaa account receive** *access-list-name* [*throttle throttle-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	aaa authorization multicast default [<i>method3</i> <i>method4</i>] Example: Router(config)# aaa authorization multicast default	Enables AAA authorization and sets parameters that restrict user access to an IPv6 multicast network.
Step 4	aaa accounting multicast default [start-stop stop-only] [broadcast] [<i>method1</i>] [<i>method2</i>] [<i>method3</i>] [<i>method4</i>] Example: Router(config)# aaa accounting multicast default	Enables AAA accounting of IPv6 multicast services for billing or security purposes when you use RADIUS.
Step 5	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 6	ipv6 multicast aaa account receive <i>access-list-name</i> [throttle <i>throttle-number</i>] Example: Router(config-if)# ipv6 multicast aaa account receive list1	Enables AAA accounting on specified groups or channels.

Disabling the Router from Receiving Unauthenticated Multicast Traffic

In some situations, access control may be needed to prevent multicast traffic from being received unless the subscriber is authenticated and the channels are authorized as per access control profiles. That is, there should be no traffic at all unless specified otherwise by access control profiles.

The following task describes how to disable the router from receiving multicast traffic to be received from unauthenticated groups or unauthorized channels.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast group-range** [*access-list-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 multicast group-range <i>[access-list-name]</i> Example: Router(config)# ipv6 multicast group-range	Disables multicast protocol actions and traffic forwarding for unauthorized groups or channels on all the interfaces in a router.

Resetting Authorization Status on an MLD Interface

The following task shows how to reset the authorization status of an interface. If no interface is specified, authorization is reset on all MLD interfaces.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 multicast aaa authorization** *[interface-type interface-number]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	clear ipv6 multicast aaa authorization <i>[interface-type interface-number]</i> Example: Router# clear ipv6 multicast aaa authorization FastEthernet 1/0	Clears parameters that restrict user access to an IPv6 multicast network.

Resetting the MLD Traffic Counters

This task explains how to reset the MLD traffic counters and verify MLD traffic information.

SUMMARY STEPS

1. **enable**

2. `clear ipv6 mld traffic`
3. `show ipv6 mld traffic`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear ipv6 mld traffic Example: Router# clear ipv6 mld traffic	Resets all MLD traffic counters.
Step 3	show ipv6 mld traffic Example: Router# show ipv6 mld traffic	Displays the MLD traffic counters.

Clearing the MLD Interface Counters

The following example shows how to clear MLD interface counters.

SUMMARY STEPS

1. `enable`
2. `clear ipv6 mld counters [interface-type]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear ipv6 mld counters [interface-type] Example: Router# clear ipv6 mld counters Ethernet1/0	Clears the MLD interface counters.

Configuring PIM

The following tasks explain how to configure PIM-SM and display PIM-SM configuration and information.

- [Configuring PIM-SM and Displaying PIM-SM Information for a Group Range, page 30](#)
- [Configuring PIM Options, page 31](#)
- [Configuring Bidirectional PIM and Displaying Bidirectional PIM Information, page 33](#)
- [Resetting the PIM Traffic Counters, page 34](#)
- [Clearing the PIM Topology Table to Reset the MRIB Connection, page 35](#)

Configuring PIM-SM and Displaying PIM-SM Information for a Group Range

This task explains how to configure PIM-SM and display PIM-SM configuration and information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim rp-address** *ipv6-address* [*group-access-list*] [**bidir**]
4. **exit**
5. **show ipv6 pim interface** [**state-on**] [**state-off**] [*type number*]
6. **show ipv6 pim group-map** [*group-name* | *group-address*] | [*group-range* | *group-mask*] [**info-source** {**bsr** | **default** | **embedded-rp** | **static**}]
7. **show ipv6 pim neighbor** [**detail**] [*interface-type interface-number* | **count**]
8. **show ipv6 pim range-list** [**config**] [*rp-address* | *rp-name*]
9. **show ipv6 pim tunnel** [*interface-type interface-number*]
10. **debug ipv6 pim** [*group-name* | *group-address* | *interface-type* | **neighbor** | **bsr**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 pim rp-address <i>ipv6-address</i> [<i>group-access-list</i>] [bidir]	Configures the address of a PIM RP for a particular group range.
	Example: Router(config)# ipv6 pim rp-address 2001:0DB8::01:800:200E:8C6C acc-grp-1	

	Command or Action	Purpose
Step 4	exit Example: Router(config-if)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 5	show ipv6 pim interface [state-on] [state-off] [<i>type number</i>] Example: Router# show ipv6 pim interface	Displays information about interfaces configured for PIM.
Step 6	show ipv6 pim group-map [<i>group-name</i> <i>group-address</i>] [<i>group-range</i> <i>group-mask</i>] [info-source { bsr default embedded-rp static }] Example: Router# show ipv6 pim group-map	Displays an IPv6 multicast group mapping table.
Step 7	show ipv6 pim neighbor [detail] [<i>interface-type interface-number</i> count] Example: Router# show ipv6 pim neighbor	Displays the PIM neighbors discovered by the Cisco IOS software.
Step 8	show ipv6 pim range-list [config] [<i>rp-address</i> <i>rp-name</i>] Example: Router# show ipv6 pim range-list	Displays information about IPv6 multicast range lists.
Step 9	show ipv6 pim tunnel [<i>interface-type interface-number</i>] Example: Router# show ipv6 pim tunnel	Displays information about the PIM register encapsulation and de-encapsulation tunnels on an interface.
Step 10	debug ipv6 pim [<i>group-name</i> <i>group-address</i> <i>interface-type</i> neighbor bsr] Example: Router# debug ipv6 pim	Enables debugging on PIM protocol activity.

Configuring PIM Options

The following task explains commands you can use to refine your configuration for PIM-SM and PIM-SSM, both in general or on specified interfaces. The task also gives various commands that can be used to verify PIM configuration and information.

SUMMARY STEPS

1. enable
2. configure terminal

3. **ipv6 pim spt-threshold infinity** [**group-list** *access-list-name*]
4. **ipv6 pim accept-register** {**list** *access-list* | **route-map** *map-name*}
5. **interface** *type number*
6. **ipv6 pim dr-priority** *value*
7. **ipv6 pim hello-interval** *seconds*
8. **ipv6 pim join-prune-interval** *seconds*
9. **exit**
10. **show ipv6 pim join-prune statistic** [*interface-type*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim spt-threshold infinity [group-list <i>access-list-name</i>] Example: Router(config)# ipv6 pim spt-threshold infinity group-list acc-grp-1	Configures when a PIM leaf router joins the SPT for the specified groups.
Step 4	ipv6 pim accept-register { list <i>access-list</i> route-map <i>map-name</i> } Example: Router(config)# ipv6 pim accept-register route-map reg-filter	Accepts or rejects registers at the RP.
Step 5	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 6	ipv6 pim dr-priority <i>value</i> Example: Router(config-if)# ipv6 pim dr-priority 3	Configures the DR priority on a PIM router.
Step 7	ipv6 pim hello-interval <i>seconds</i> Example: Router(config-if)# ipv6 pim hello-interval 45	Configures the frequency of PIM hello messages on an interface.

	Command or Action	Purpose
Step 8	ipv6 pim join-prune-interval <i>seconds</i> Example: Router(config-if)# ipv6 pim join-prune-interval 75	Configures periodic join and prune announcement intervals for a specified interface.
Step 9	exit Example: Router(config-if)# exit	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 10	show ipv6 pim join-prune statistic [<i>interface-type</i>] Example: Router# show ipv6 pim join-prune statistic	Displays the average join-prune aggregation for the most recently aggregated 1000, 10,000, and 50,000 packets for each interface.

Configuring Bidirectional PIM and Displaying Bidirectional PIM Information

This task explains how to configure bidirectional PIM and use **show** commands to display bidirectional PIM information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim rp-address** *ipv6-address* [*group-access-list*] [**bidir**]
4. **exit**
5. **show ipv6 pim df** [*interface-type interface-number*] [*rp-address*]
6. **show ipv6 pim df winner** [*interface-type interface-number*] [*rp-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ipv6 pim rp-address <i>ipv6-address</i> <i>[group-access-list] [bidir]</i> Example: Router(config)# ipv6 pim rp-address 2001:0DB8::01:800:200E:8C6C bidir	Configures the address of a PIM RP for a particular group range. Use of the bidir keyword means that the group range will be used for bidirectional shared-tree forwarding.
Step 4	exit Example: Router(config-if)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 5	show ipv6 pim df [<i>interface-type</i> <i>interface-number</i>] [<i>rp-address</i>] Example: Router# show ipv6 pim df	Displays the designated forwarder (DF)-election state of each interface for RP.
Step 6	show ipv6 pim df winner [<i>interface-type</i> <i>interface-number</i>] [<i>rp-address</i>] Example: Router# show ipv6 pim df winner ethernet 1/0 200::1	Displays the DF-election winner on each interface for each RP.

Resetting the PIM Traffic Counters

If PIM malfunctions or in order to verify that the expected number of PIM packets are received and sent, the user can clear PIM traffic counters. Once the traffic counters are cleared, the user can enter the **show ipv6 pim traffic** command to verify that PIM is functioning correctly and that PIM packets are being received and sent correctly.

This task explains how to reset the PIM traffic counters and verify PIM traffic information.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim counters**
3. **show ipv6 pim traffic**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear ipv6 pim counters Example: Router# clear ipv6 pim counters	Resets the PIM traffic counters.
Step 3	show ipv6 pim traffic Example: Router# show ipv6 pim traffic	Displays the PIM traffic counters.

Clearing the PIM Topology Table to Reset the MRIB Connection

No configuration is necessary to use the MRIB. However, users may in certain situations want to clear the PIM topology table in order to reset the MRIB connection, and verify MRIB information.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 pim topology** [*group-name* | *group-address*]
3. **show ipv6 mrib client** [**filter**] [**name** {*client-name* | *client-name:client-id*}]
4. **show ipv6 mrib route** [**link-local** | **summary** | *source-address* | *source-name* | *] [*group-name* | *group-address* [*prefix-length*]]
5. **show ipv6 pim topology** [**link-local** | **route-count** | *group-name* | *group-address*] [*source-address* | *source-name*]
6. **debug ipv6 mrib client**
7. **debug ipv6 mrib io**
8. **debug ipv6 mrib proxy**
9. **debug ipv6 mrib route** [*group-name* | *group-address*]
10. **debug ipv6 mrib table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear ipv6 pim topology [<i>group-name</i> <i>group-address</i>] Example: Router# clear ipv6 pim topology FF04::10	Clears the PIM topology table.
Step 3	show ipv6 mrib client [filter] [name { <i>client-name</i> <i>client-name:client-id</i> }] Example: Router# show ipv6 mrib client	Displays multicast-related information about an interface.
Step 4	show ipv6 mrib route [link-local summary <i>source-address</i> <i>source-name</i> *] [<i>group-name</i> <i>group-address</i> [<i>prefix-length</i>]] Example: Router# show ipv6 mrib route	Displays the MRIB route information.
Step 5	show ipv6 pim topology [link-local route-count <i>group-name</i> <i>group-address</i>] [<i>source-address</i> <i>source-name</i>] Example: Router# show ipv6 pim topology	Displays PIM topology table information for a specific group or all groups.
Step 6	debug ipv6 mrib client Example: Router# debug ipv6 mrib client	Enables debugging on MRIB client management activity.
Step 7	debug ipv6 mrib io Example: Router# debug ipv6 mrib io	Enables debugging on MRIB I/O events.
Step 8	debug ipv6 mrib proxy Example: Router# debug ipv6 mrib proxy	Enables debugging on MRIB proxy activity between the route processor and line cards on distributed router platforms.

	Command or Action	Purpose
Step 9	debug ipv6 mrib route [<i>group-name</i> <i>group-address</i>] Example: Router# debug ipv6 mrib route	Displays information about MRIB routing entry-related activity.
Step 10	debug ipv6 mrib table Example: Router# debug ipv6 mrib table	Enables debugging on MRIB table management activity.

Configuring a BSR

The following tasks explain how to perform BSR configuration and to verify BSR configuration and information:

- [Configuring a BSR and Verifying BSR Information, page 37](#)
- [Sending PIM RP Advertisements to the BSR, page 38](#)
- [Configuring BSR for Use Within Scoped Zones, page 39](#)
- [Configuring BSR Routers to Announce Scope-to-RP Mappings, page 40](#)

Configuring a BSR and Verifying BSR Information

This task describes how to configure a BSR on a specified interface and verify BSR configuration information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*]
4. **interface** *type number*
5. **ipv6 pim bsr border**
6. **exit**
7. **show ipv6 pim bsr** {**election** | **rp-cache** | **candidate-rp**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr candidate bsr <i>ipv6-address</i> [<i>hash-mask-length</i>] [priority <i>priority-value</i>] Example: Router(config)# ipv6 pim bsr candidate bsr 2001:0DB8:3000:3000::42 124 priority 10	Configures a router to be a candidate BSR.
Step 4	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	ipv6 pim bsr border Example: Router(config-if)# ipv6 pim bsr border	Configures a border for all BSMs of any scope on a specified interface.
Step 6	exit Example: Router(config-if)# exit	Enter this command twice to exit interface configuration mode and enter privileged EXEC mode.
Step 7	show ipv6 pim bsr { election rp-cache candidate-rp } Example: Router# show ipv6 pim bsr election	Displays information related to PIM BSR protocol processing.

Sending PIM RP Advertisements to the BSR

This task explains how to configure a router to send PIM RP advertisements to the BSR.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]
- interface** *type number*

5. ipv6 pim bsr border

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr candidate rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [interval <i>seconds</i>] [scope <i>scope-value</i>] [bidir] Example: Router(config)# ipv6 pim bsr candidate rp 2001:0DB8:3000:3000::42 priority 0	Sends PIM RP advertisements to the BSR.
Step 4	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	ipv6 pim bsr border Example: Router(config-if)# ipv6 pim bsr border	Configures a border for all BSMs of any scope on a specified interface.

Configuring BSR for Use Within Scoped Zones

The following task enables you to use BSR within scoped zones. A user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the domain.

If scope is specified on the candidate RP, then this router will advertise itself as C-RP only to the BSR for the specified scope. If the group list is specified along with the scope, then only prefixes in the access list with the same scope as that configured will be advertised.

If a scope is specified on the bootstrap router, the BSR will originate BSMs including the group range associated with the scope and accept C-RP announcements for groups that belong to the given scope.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim bsr candidate bsr** *ipv6-address* [*hash-mask-length*] [**priority** *priority-value*]

4. **ipv6 pim bsr candidate rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**interval** *seconds*] [**scope** *scope-value*] [**bidir**]
5. **interface** *type number*
6. **ipv6 multicast boundary scope** *scope-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr candidate bsr <i>ipv6-address</i> [<i>hash-mask-length</i>] [priority <i>priority-value</i>] Example: Router(config)# ipv6 pim bsr candidate bsr 2001:0DB8:1:1:4	Configures a router to be a candidate BSR.
Step 4	ipv6 pim bsr candidate rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [interval <i>seconds</i>] [scope <i>scope-value</i>] [bidir] Example: Router(config)# ipv6 pim bsr candidate rp 2001:0DB8:1:1:1 group-list list scope 6	Configures the candidate RP to send PIM RP advertisements to the BSR.
Step 5	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 6	ipv6 multicast boundary scope <i>scope-value</i> Example: Router(config-if)# ipv6 multicast boundary scope 6	Configures a multicast boundary on the interface for a specified scope.

Configuring BSR Routers to Announce Scope-to-RP Mappings

IPv6 BSR routers can be statically configured to announce scope-to-RP mappings directly instead of learning them from candidate-RP messages. A user might want to configure a BSR router to announce scope-to-RP mappings so that an RP that does not support BSR is imported into the BSR. Enabling this feature also allows an RP positioned outside the enterprise's BSR domain to be learned by the known remote RP on the local candidate BSR routers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 pim bsr announced rp** *ipv6-address* [**group-list** *access-list-name*] [**priority** *priority-value*] [**bidir**] [**scope** *scope-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 pim bsr announced rp <i>ipv6-address</i> [group-list <i>access-list-name</i>] [priority <i>priority-value</i>] [bidir] [scope <i>scope-value</i>] Example: Router(config)# ipv6 pim bsr announced rp 2001:0DB8:3000:3000::42 priority 0	Announces scope-to-RP mappings directly from the BSR for the specified candidate RP.

Configuring SSM Mapping

When the SSM mapping feature is enabled, DNS-based SSM mapping is automatically enabled, which means that the router will look up the source of a multicast MLD version 1 report from a DNS server.

You can use either DNS-based or static SSM mapping, depending on your router configuration. If you choose to use static SSM mapping, you can configure multiple static SSM mappings. If multiple static SSM mappings are configured, the source addresses of all matching access lists will be used.

This task explains how to enable SSM mapping, disable DNS-based mapping, and configure static SSM mapping.

Restrictions

To use DNS-based SSM mapping, the router needs to find at least one correctly configured DNS server, to which the router may be directly attached.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mld** [**vrf** *vrf-name*] **ssm-map enable**

4. **no ipv6 mld** [*vrf vrf-name*] **ssm-map query dns**
5. **ipv6 mld ssm-map** [*vrf vrf-name*] **static** *access-list source-address*
6. **exit**
7. **show ipv6 mld ssm-map** [*source-address*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mld [<i>vrf vrf-name</i>] ssm-map enable Example: Router(config)# ipv6 mld ssm-map enable	Enables the SSM mapping feature for groups in the configured SSM range.
Step 4	no ipv6 mld [<i>vrf vrf-name</i>] ssm-map query dns Example: Router(config)# no ipv6 mld ssm-map query dns	Disables DNS-based SSM mapping.
Step 5	ipv6 mld ssm-map [<i>vrf vrf-name</i>] static <i>access-list source-address</i> Example: Router(config)# ipv6 mld ssm-map static SSM_MAP_ACL_2 2001:0DB8:1::1	Configures static SSM mappings.
Step 6	exit Example: Router(config-if)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 7	show ipv6 mld ssm-map [<i>source-address</i>] Example: Router# show ipv6 mld ssm-map	Displays SSM mapping information.

Configuring Static Mroutes

This task explains how to configure a static multicast route and verify static mroute information. Static multicast routes (mroutes) in IPv6 can be implemented as an extension of IPv6 static routes. You can configure your router to use a static route for unicast routing only, to use a static multicast route for multicast RPF selection only, or to use a static route for both unicast routing and multicast RPF selection.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix/prefix-length* { *ipv6-address* | *interface-type interface-number* [*ipv6-address*] } [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*tag tag*]
4. **exit**
5. **show ipv6 mroute** [**link-local** | [*group-name* | *group-address* [*source-address* | *source-name*]]] [**summary**] [**count**]
6. **show ipv6 mroute** [**link-local** | *group-name* | *group-address*] **active** [*kbps*]
7. **show ipv6 rpf** *ipv6-prefix*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>] } [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>tag tag</i>] Example: Router(config)# ipv6 route 2001:0DB8::/64 6::6 100	Establishes static IPv6 routes. The example shows a static route used for both unicast routing and multicast RPF selection.
Step 4	exit Example: Router(config-if)# exit	Exits global configuration mode, and returns the router to privileged EXEC mode.
Step 5	show ipv6 mroute [link-local [<i>group-name</i> <i>group-address</i> [<i>source-address</i> <i>source-name</i>]]] [summary] [count] Example: Router# show ipv6 mroute ff07::1	Displays the contents of the IPv6 multicast routing table.

	Command or Action	Purpose
Step 6	show ipv6 mroute [link-local <i>group-name</i> <i>group-address</i>] active [<i>kbits</i>] Example: Router# show ipv6 mroute active	Displays the active multicast streams on the router.
Step 7	show ipv6 rpf <i>ipv6-prefix</i> Example: Router# show ipv6 rpf 2001:0DB8::1:1:2	Checks RPF information for a given unicast host address and prefix.

Configuring IPv6 Multiprotocol BGP

The following tasks explain how to configure IPv6 multiprotocol BGP to perform multicast routing. Note that these multicast BGP tasks related to IPv6 multicast are similar to those multicast BGP tasks for IPv6 unicast.

- [Configuring an IPv6 Peer Group to Perform Multicast BGP Routing, page 44](#)
- [Advertising Routes into IPv6 Multiprotocol BGP, page 46](#)
- [Redistributing Prefixes into IPv6 Multiprotocol BGP, page 47](#)
- [Assigning a BGP Administrative Distance, page 49](#)
- [Generating Translate Updates for IPv6 Multicast BGP, page 50](#)
- [Resetting BGP Sessions, page 51](#)
- [Clearing External BGP Peers, page 52](#)
- [Clearing IPv6 BGP Route Dampening Information, page 53](#)
- [Clearing IPv6 BGP Flap Statistics, page 53](#)

Configuring an IPv6 Peer Group to Perform Multicast BGP Routing

The following tasks explain how to configure an IPv6 peer group to perform multicast BGP routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **neighbor** *peer-group-name* **peer-group**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **address-family** **ipv6** [**unicast** | **multicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address*} **peer-group** *peer-group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp as-number Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	neighbor peer-group-name peer-group Example: Router(config-router)# neighbor group1 peer-group	Creates an multicast BGP peer group.
Step 5	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 2001:0DB8:0:CC00::1 remote-as 64600	Adds the IPv6 address of the neighbor in the specified autonomous system to the IPv6 multicast BGP neighbor table of the local router. <ul style="list-style-type: none"> The <i>ipv6-address</i> argument in the neighbor remote-as command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
Step 6	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6 multicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.

	Command or Action	Purpose
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 2001:0DB8:0:CC00::1 activate	Enables the neighbor to exchange prefixes for the specified family type with the neighbor and the local router. <ul style="list-style-type: none"> To avoid extra configuration steps for each neighbor, use the neighbor activate command with the <i>peer-group-name</i> argument as an alternative in this step.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> } peer-group <i>peer-group-name</i> Example: Router(config-router-af)# neighbor 2001:0DB8:0:CC00::1 peer-group group1	Assigns the IPv6 address of a BGP neighbor to a peer group.

What to Do Next

Refer to “Configuring an IPv6 Multiprotocol BGP Peer Group” in the [Implementing Multiprotocol BGP for IPv6](#) document and the “BGP Features Roadmap” chapter in the [Cisco IOS IP Routing Configuration Guide](#) for more information on assigning options to peer groups and making a BGP or multicast BGP neighbor a member of a peer group.

Advertising Routes into IPv6 Multiprotocol BGP

This task explains how to advertise (inject) a prefix into IPv6 multicast BGP. Note that this task and other multicast BGP tasks related to IPv6 multicast are similar to those multicast BGP tasks for IPv6 unicast.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **network** *ipv6-address/prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router <i>bgp as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6 multicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command. The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	network <i>ipv6-address/prefix-length</i> Example: Router(config-router-af)# network 2001:0DB8::/24	Advertises (injects) the specified prefix into the IPv6 BGP database. (The routes must first be found in the IPv6 unicast routing table.) <ul style="list-style-type: none"> Specifically, the prefix is injected into the database for the address family specified in the previous step. Routes are tagged from the specified prefix as “local origin.” The <i>ipv6-prefix</i> argument in the network command must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The <i>prefix-length</i> argument is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.

What to Do Next

Refer to the section “Advertising Routes into IPv6 Multiprotocol BGP” in the [Implementing Multiprotocol BGP for IPv6](#) implementation guide for more information on assigning options to peer groups and making a BGP or multicast BGP neighbor a member of a peer group.

Redistributing Prefixes into IPv6 Multiprotocol BGP

This task explains how to redistribute (inject) prefixes from another routing protocol into IPv6 multicast BGP. Note that this task and other multicast BGP tasks related to IPv6 multicast are similar to those multicast BGP tasks for IPv6 unicast.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **router** **bgp** *as-number*
4. **address-family** **ipv6** [**unicast** | **multicast**]
5. **redistribute** *protocol* [*process-id*] [**level-1** | **level-1-2** | **level-2**] [**metric** *metric-value*] [**metric-type** {**internal** | **external**}] [**route-map** *map-name*]
6. **exit**
7. **debug** **bgp** **ipv6** {**unicast** | **multicast**} **dampening** [**prefix-list** *prefix-list-name*]
8. **debug** **bgp** **ipv6** {**unicast** | **multicast**} **updates** [*ipv6-address*] [**prefix-list** *prefix-list-name*] [**in** | **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified BGP routing process.
Step 4	address-family ipv6 { unicast multicast } Example: Router(config-router)# address-family ipv6 multicast	Specifies the IPv6 address family, and enters address family configuration mode. <ul style="list-style-type: none">• The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.• The multicast keyword specifies IPv6 multicast address prefixes.
Step 5	redistribute <i>protocol</i> [<i>process-id</i>] [level-1 level-1-2 level-2] [metric <i>metric-value</i>] [metric-type { internal external }] [route-map <i>map-name</i>] Example: Router(config-router-af)# redistribute rip	Specifies the routing protocol from which prefixes should be redistributed into IPv6 multicast BGP. <ul style="list-style-type: none">• The <i>protocol</i> argument can be one of the following keywords: bgp, connected, isis, rip, or static. Note The connected keyword refers to routes that are established automatically by IPv6 having been enabled on an interface.
Step 6	exit Example: Router(config-router-af)# exit	Enter this command three times to exit address family configuration mode and enter privileged EXEC mode.

	Command or Action	Purpose
Step 7	<pre>debug bgp ipv6 {unicast multicast} dampening [prefix-list <i>prefix-list-name</i>]</pre> <p>Example: Router# debug bgp ipv6 multicast</p>	Displays debugging messages for IPv6 BGP dampening.
Step 8	<pre>debug bgp ipv6 {unicast multicast} updates [<i>ipv6-address</i>] [prefix-list <i>prefix-list-name</i>] [in out]</pre> <p>Example: Router# debug bgp ipv6 multicast updates</p>	Displays debugging messages for IPv6 BGP update packets.

What to Do Next

Refer to the section “Redistributing Prefixes into IPv6 Multiprotocol BGP” in the [Implementing Multiprotocol BGP for IPv6](#) implementation guide for more information on assigning options to peer groups and making a BGP or multicast BGP neighbor a member of a peer group.

Assigning a BGP Administrative Distance

This task explains how to specify an administrative distance for multicast BGP routes to be used in RPF lookups for comparison with unicast routes. Please note that this task and other multicast BGP tasks related to IPv6 multicast are similar to those multicast BGP tasks for IPv6 unicast.



Caution

Changing the administrative distance of BGP internal routes is considered dangerous and is not recommended. One problem that can arise is the accumulation of routing table inconsistencies, which can break routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **distance bgp** *external-distance internal-distance local-distance*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 100	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6 multicast	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	distance bgp <i>external-distance</i> <i>internal-distance</i> <i>local-distance</i> Example: Router(config-router)# distance bgp 20 20 200	Assigns a BGP administrative distance.

Generating Translate Updates for IPv6 Multicast BGP

This task explains how to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates received from a peer.

The multicast BGP translate-update feature generally is used in an multicast BGP-capable router that peers with a customer site that has only a BGP-capable router; the customer site has not or cannot upgrade its router to an multicast BGP-capable image. Because the customer site cannot originate multicast BGP advertisements, the router with which it peers will translate the BGP prefixes into multicast BGP prefixes, which are used for multicast-source RPF lookup.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **address-family ipv6** [**unicast** | **multicast**]
5. **neighbor** *ipv6-address* **translate-update ipv6 multicast** [**unicast**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 100	Enters router configuration mode for the specified routing process.
Step 4	address-family ipv6 [unicast multicast] Example: Router(config-router)# address-family ipv6 multicast	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 5	neighbor <i>ipv6-address</i> translate-update ipv6 multicast [unicast] Example: Router(config-router)# neighbor 2001:0DB8:7000::2 translate-update ipv6 multicast	Generates multiprotocol IPv6 BGP updates that correspond to unicast IPv6 updates received from a peer.

Resetting BGP Sessions

The following shows how to reset IPv6 BGP sessions.

SUMMARY STEPS

- enable**
- clear bgp ipv6** {**unicast** | **multicast**} {***** | *autonomous-system-number* | *ip-address* | *ipv6-address* | *peer-group-name*} [**soft**] [**in** | **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} {* autonomous-system-number ip-address ipv6-address peer-group-name} [soft] [in out] Example: Router# clear bgp ipv6 unicast peer-group marketing soft out	Resets IPv6 BGP sessions.

Clearing External BGP Peers

The following example shows how to clear external BGP peers and members of an IPv6 BGP peer group.

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6 {unicast | multicast} external [soft] [in | out]**
3. **clear bgp ipv6 {unicast | multicast} peer-group [name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} external [soft] [in out] Example: Router# clear bgp ipv6 unicast external soft in	Clears external IPv6 BGP peers.
Step 3	clear bgp ipv6 {unicast multicast} peer-group [name] Example: Router# clear bgp ipv6 unicast peer-group	Clears all members of an IPv6 BGP peer group.

Clearing IPv6 BGP Route Dampening Information

This task explains how to clear IPv6 BGP route dampening information and how to unsuppress suppressed routes.

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6 {unicast | multicast} dampening [ipv6-prefix/prefix-length]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} dampening [ipv6-prefix/prefix-length] Example: Router# clear bgp ipv6 unicast dampening 2001:0DB8:7000::/64	Clears IPv6 BGP route dampening information and unsuppress the suppressed routes.

Clearing IPv6 BGP Flap Statistics

The following example shows how to clear IPv6 BGP flap statistics.

SUMMARY STEPS

1. **enable**
2. **clear bgp ipv6 {unicast | multicast} flap-statistics [ipv6-prefix/prefix-length | regexp regexp | filter-list list]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear bgp ipv6 {unicast multicast} flap-statistics [ipv6-prefix/prefix-length regexp regexp filter-list list] Example: Router# clear bgp ipv6 multicast flap-statistics	Clears IPv6 BGP flap statistics.

Using MFIB in IPv6 Multicast

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled. The following tasks explain how to display information to verify MFIB configuration and operation and reset MFIB as needed.

- [Verifying MFIB Operation in IPv6 Multicast, page 54](#)
- [Resetting MFIB Traffic Counters, page 55](#)
- [Disabling Default Features in IPv6 Multicast, page 56](#)

Verifying MFIB Operation in IPv6 Multicast

Use the following **show** commands and **debug** command as needed to display, verify, and troubleshoot MFIB use in IPv6 multicast.

SUMMARY STEPS

1. **enable**
2. **show ipv6 mfib** [**link-local** | *ipv6-prefix/prefix-length* | *group-name* | *group-address* [*source-name* | *source-address*]] [**verbose**]
3. **show ipv6 mfib** [**link-local** | *group-name* | *group-address*] **active** [*kbps*]
4. **show ipv6 mfib** [**link-local** | *group-name* | *group-address* [*source-name* | *source-address*]] **count**
5. **show ipv6 mfib interface**
6. **show ipv6 mfib status**
7. **show ipv6 mfib summary**
8. **debug ipv6 mfib** [*group-name* | *group-address*] [**adjacency** | **signal** | **db** | **init** | **mrrib** | **pak** | **ps**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	show ipv6 mfib [link-local <i>ipv6-prefix/prefix-length</i> <i>group-name</i> <i>group-address</i> [<i>source-name</i> <i>source-address</i>]] [verbose] Example: Router# show ipv6 mfib	Displays the forwarding entries and interfaces in the IPv6 MFIB.
Step 3	show ipv6 mfib [link-local <i>group-name</i> <i>group-address</i>] active [<i>kbps</i>] Example: Router# show ipv6 mfib active	Displays the rate at which active sources are sending to multicast groups.

	Command or Action	Purpose
Step 4	show ipv6 mfib [link-local <i>group-name</i> <i>group-address</i> [<i>source-name</i> <i>source-address</i>]] count Example: Router# show ipv6 mfib count	Displays summary traffic statistics from the MFIB about the group and source.
Step 5	show ipv6 mfib interface Example: Router# show ipv6 mfib interface	Displays information about IPv6 multicast-enabled interfaces and their forwarding status.
Step 6	show ipv6 mfib status Example: Router# show ipv6 mfib status	Displays general MFIB configuration and operational status.
Step 7	show ipv6 mfib summary Example: Router# show ipv6 mfib summary	Displays summary information about the number of IPv6 MFIB entries and interfaces.
Step 8	debug ipv6 mfib [<i>group-name</i> <i>group-address</i>] [adjacency signal db init mrrib pak ps] Example: Router# debug ipv6 mfib FF04::10 pak	Enables debugging output on the IPv6 MFIB.

Resetting MFIB Traffic Counters

The following example shows how to reset all active MFIB traffic counters.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 mfib counters** [*group-name* | *group-address* [*source-address* | *source-name*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear ipv6 mfib counters [<i>group-name</i> <i>group-address</i> [<i>source-address</i> <i>source-name</i>]] Example: Router# clear ipv6 mfib counters FF04::10	Resets all active MFIB traffic counters.

Disabling Default Features in IPv6 Multicast

Several features are automatically enabled when IPv6 multicast is used. However, a user may want to disable certain features in response to certain situations. The following tasks describe these situations and how to disable specific IPv6 multicast features.

- [Disabling Embedded RP Support in IPv6 PIM, page 56](#)
- [Turning Off IPv6 PIM on a Specified Interface, page 57](#)
- [Disabling MLD Router-Side Processing, page 58](#)
- [Disabling MFIB on the Router, page 58](#)
- [Disabling MFIB on a Distributed Platform, page 59](#)
- [Disabling MFIB Interrupt-Level IPv6 Multicast Forwarding, page 60](#)

Disabling Embedded RP Support in IPv6 PIM

A user might want to disable embedded RP support on an interface if all of the routers in the domain do not support embedded RP. This task explains how to disable embedded RP support in IPv6 PIM.



Note

This task disables PIM completely, not just embedded RP support in IPv6 PIM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ipv6 pim rp embedded**
4. **interface** *type number*
5. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	no ipv6 pim rp embedded Example: Router(config)# no ipv6 pim rp embedded	Disables embedded RP support in IPv6 PIM.

	Command or Action	Purpose
Step 4	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	no ipv6 pim Example: Router(config-if)# no ipv6 pim	Turns off IPv6 PIM on a specified interface.

Turning Off IPv6 PIM on a Specified Interface

A user might only want specified interfaces to perform IPv6 multicast and will therefore want to turn off PIM on a specified interface. This task explains how to turn off PIM on a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 pim**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	no ipv6 pim Example: Router(config-if)# no ipv6 pim	Turns off IPv6 PIM on a specified interface.

Disabling MLD Router-Side Processing

A user might only want specified interfaces to perform IPv6 multicast and will therefore want to turn off MLD router-side processing on a specified interface. Use the following task to disable MLD router-side processing on a specified interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 mld router**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	no ipv6 mld router Example: Router(config-if)# no ipv6 mld router	Disables MLD router-side processing on a specified interface.

Disabling MFIB on the Router

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled. However, a user may want to disable multicast forwarding on the router. The following task explains how to disable multicast forwarding on the router:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ipv6 mfib**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	no ipv6 mfib Example: Router(config)# no ipv6 mfib	Disables IPv6 multicast forwarding on the router.

Disabling MFIB on a Distributed Platform

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled. However, a user may want to disable multicast forwarding on a distributed platform. The following task explains how to disable multicast forwarding on a distributed platform:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 mfib-mode centralized-only**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 mfib-mode centralized-only Example: Router(config)# ipv6 mfib-mode centralized-only	Disables distributed forwarding on a distributed platform.

Disabling MFIB Interrupt-Level IPv6 Multicast Forwarding

MFIB interrupt-level IPv6 multicast forwarding of outgoing packets on a specific interface is enabled on interfaces that support Cisco Express Forwarding. However, a user may want to disable MFIB interrupt-level forwarding on a specified interface. The following task explains how to disable this feature:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no ipv6 mfib cef output**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	no ipv6 mfib cef output Example: Router(config-if)# no ipv6 mfib cef output	Disables MFIB interrupt-level IPv6 multicast forwarding of outgoing packets on a specific interface.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 mfib Command, page 61](#)
- [Sample Output from the show ipv6 mfib active Command, page 62](#)
- [Sample Output from the show ipv6 mfib count Command, page 62](#)
- [Sample Output from the show ipv6 mfib interface Command, page 62](#)
- [Sample Output from the show ipv6 mfib summary Command, page 62](#)
- [Sample Output from the show ipv6 mld groups Command, page 63](#)
- [Sample Output from the show ipv6 mld groups summary Command, page 64](#)

- [Sample Output from the show ipv6 mld interface Command, page 64](#)
- [Sample Output from the show ipv6 mld ssm-map Command, page 64](#)
- [Sample Output from the show ipv6 mld traffic Command, page 64](#)
- [Sample Output from the show ipv6 mrrib client Command, page 65](#)
- [Sample Output from the show ipv6 mrrib route Command, page 65](#)
- [Sample Output from the show ipv6 mroute Command, page 65](#)
- [Sample Output from the show ipv6 mroute active Command, page 65](#)
- [Sample Output from the show ipv6 pim bsr Command, page 66](#)
- [Sample Output from the show ipv6 pim group-map Command, page 66](#)
- [Sample Output from the show ipv6 pim interface Command, page 66](#)
- [Sample Output from the show ipv6 pim join-prune statistic Command, page 66](#)
- [Sample Output from the show ipv6 pim neighbor Command, page 67](#)
- [Sample Output from the show ipv6 pim range-list Command, page 67](#)
- [Sample Output from the show ipv6 pim topology Command, page 67](#)
- [Sample Output from the show ipv6 pim traffic Command, page 68](#)
- [Sample Output from the show ipv6 pim tunnel Command, page 68](#)
- [Sample Output from the show ipv6 rpf Command, page 69](#)

Sample Output from the show ipv6 mfib Command

The following example displays the forwarding entries and interfaces in the MFIB. The router is configured for fast switching, and it has a receiver joined to FF05::1 on Ethernet1/1 and a source (2001:0DB8:1:1:20) sending on Ethernet1/2:

Router# show ipv6 mfib

```
IP Multicast Forwarding Information Base
Entry Flags: C - Directly Connected, S - Signal, IA - Inherit A flag,
              AR - Activity Required, D - Drop
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kbits per second
Other counts: Total/RPF failed/Other drops
Interface Flags: A - Accept, F - Forward, NS - Negate Signalling
                  IC - Internal Copy, NP - Not platform switched
                  SP - Signal Present
Interface Counts: FS Pkt Count/PS Pkt Count
(*,FF00::/8) Flags: C
  Forwarding: 0/0/0/0, Other: 0/0/0
  Tunnel0 Flags: NS
(*,FF00::/15) Flags: D
  Forwarding: 0/0/0/0, Other: 0/0/0
(*,FF05::1) Flags: C
  Forwarding: 2/0/100/0, Other: 0/0/0
  Tunnel0 Flags: A NS
  Ethernet1/1 Flags: F NS
    Pkts: 0/2
(2001:0DB8:1:1:20,FF05::1) Flags:
  Forwarding: 5/0/100/0, Other: 0/0/0
  Ethernet1/2 Flags: A
  Ethernet1/1 Flags: F NS
    Pkts: 3/2
(*,FF10::/15) Flags: D
  Forwarding: 0/0/0/0, Other: 0/0/0
```

Sample Output from the show ipv6 mfib active Command

The following example displays statistics on the rate at which active IP multicast sources are sending information. The router is switching traffic from 2001:0DB8:1:1:200 to FF05::1:

```
Router# show ipv6 mfib active

Active IPv6 Multicast Sources - sending >= 4 kbps
Group: FF05::1
  Source: 2001:0DB8:1:1:200
    Rate: 20 pps/16 kbps(1sec), 0 kbps(last 128 sec)
```

Sample Output from the show ipv6 mfib count Command

The following example displays statistics from the MFIB about the group and source. The router is switching traffic from 2001:0DB8:1:1:200 to FF05::1:

```
Router# show ipv6 mfib count

IP Multicast Statistics
54 routes, 7 groups, 0.14 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)
Group: FF00::/8
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
Group: FF00::/15
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
Group: FF05::1
  RP-tree: Forwarding: 2/0/100/0, Other: 0/0/0
    Source: 10::1:1:200, Forwarding: 367/10/100/7, Other: 0/0/0
    Tot. shown: Source count: 1, pkt count: 369
Group: FF10::/15
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
Group: FF20::/15
  RP-tree: Forwarding: 0/0/0/0, Other: 0/0/0
```

Sample Output from the show ipv6 mfib interface Command

The following example displays information about IPv6 multicast-enabled interfaces and their forwarding status. The router is configured for fast switching:

```
Router# show ipv6 mfib interface

IPv6 Multicast Forwarding (MFIB) status:
  Configuration Status: enabled
  Operational Status: running

MFIB interface      status      CEF-based output
                  [configured,available]
Ethernet1/1         up         [yes      ,yes    ]
Ethernet1/2         up         [yes      ,?      ]
Tunnel0             up         [yes      ,?      ]
Tunnel1            up         [yes      ,?      ]
```

Sample Output from the show ipv6 mfib summary Command

The following example displays summary information about the number of IPv6 MFIB entries and interfaces:

```
Router# show ipv6 mfib summary

IPv6 MFIB summary:
  54      total entries [1 (S,G), 7 (*,G), 46 (*,G/m)]
  17      total MFIB interfaces
```

Sample Output from the show ipv6 mld groups Command

The following is sample output from the **show ipv6 mld groups** command. It shows all of the groups joined by Fast Ethernet interface 2/1, including link-local groups used by network protocols.

Router# **show ipv6 mld groups FastEthernet 2/1**

MLD Connected Group Membership			
Group Address	Interface	Uptime	Expires
FF02::2	FastEthernet2/1	3d18h	never
FF02::D	FastEthernet2/1	3d18h	never
FF02::16	FastEthernet2/1	3d18h	never
FF02::1:FF00:1	FastEthernet2/1	3d18h	00:00:27
FF02::1:FF00:79	FastEthernet2/1	3d18h	never
FF02::1:FF23:83C2	FastEthernet2/1	3d18h	00:00:22
FF02::1:FFAF:2C39	FastEthernet2/1	3d18h	never
FF06:7777::1	FastEthernet2/1	3d18h	00:00:26

Sample Output from the show ipv6 mld groups summary Command

The following is sample output from the **show ipv6 mld groups summary** command:

```
Router# show ipv6 mld groups summary
```

```
MLD Route Summary
  No. of (*,G) routes = 5
  No. of (S,G) routes = 0
```

Sample Output from the show ipv6 mld interface Command

The following is sample output from the **show ipv6 mld interface** command for Fast Ethernet interface 2/1:

```
Router# show ipv6 mld interface FastEthernet 2/1
```

```
FastEthernet2/1 is up, line protocol is up
Internet address is FE80::205:5FFF:FEAF:2C39/10
MLD is enabled in interface
Current MLD version is 2
MLD query interval is 125 seconds
MLD querier timeout is 255 seconds
MLD max query response time is 10 seconds
Last member query response interval is 1 seconds
MLD activity: 25 joins, 17 leaves
MLD querying router is FE80::205:5FFF:FEAF:2C39 (this system)
```

Sample Output from the show ipv6 mld ssm-map Command

The following examples show SSM mapping for the source address 2001:0DB8::1:

```
Router# show ipv6 mld ssm-map 2001:0DB8::1
```

```
Group address   : 2001:0DB8::1
Group mode ssm  : TRUE
Database        : STATIC
Source list     : 2001:0DB8::2
                  2001:0DB8::3
```

```
Router# show ipv6 mld ssm-map 2001:0DB8::2
```

```
Group address   : 2001:0DB8::2
Group mode ssm  : TRUE
Database        : DNS
Source list     : 2001:0DB8::3
                  2001:0DB8::1
```

Sample Output from the show ipv6 mld traffic Command

The following example displays the MLD protocol messages received and sent.

```
Router# show ipv6 mld traffic
```

```
MLD Traffic Counters
Elapsed time since counters cleared:00:00:21
```

	Received	Sent
Valid MLD Packets	3	1
Queries	1	0
Reports	2	1
Leaves	0	0
Mtrace packets	0	0
Errors:		
Malformed Packets		0

```

Bad Checksums                                0
Martian source                               0
Packets Received on MLD-disabled Interface  0

```

Sample Output from the show ipv6 mrib client Command

The following is sample output from the **show ipv6 mrib client** command:

```

Router# show ipv6 mrib client

IP MRIB client-connections
igmp:145          (connection id 0)
pim:146 (connection id 1)
mfib ipv6:3       (connection id 2)
slot 3  mfib ipv6 rp agent:16  (connection id 3)
slot 1  mfib ipv6 rp agent:16  (connection id 4)
slot 0  mfib ipv6 rp agent:16  (connection id 5)
slot 4  mfib ipv6 rp agent:16  (connection id 6)
slot 2  mfib ipv6 rp agent:16  (connection id 7)

```

Sample Output from the show ipv6 mrib route Command

The following is sample output from the **show ipv6 mrib route** command using the **summary** keyword:

```

Router# show ipv6 mrib route summary

MRIB Route-DB Summary
  No. of (*,G) routes = 52
  No. of (S,G) routes = 0
  No. of Route x Interfaces (RxI) = 10

```

Sample Output from the show ipv6 mroute Command

Using the **show ipv6 mroute** command is a good way to dynamically verify that multicast IPv6 data is flowing. The following is sample output from the **show ipv6 mroute** command:

```

Router# show ipv6 mroute ff07::1

Multicast Routing Table
Flags:D - Dense, S - Sparse, B - Bidir Group, s - SSM Group,
      C - Connected, L - Local, I - Received Source Specific Host Report,
      P - Pruned, R - RP-bit set, F - Register flag, T - SPT-bit set,
      J - Join SPT
Timers:Uptime/Expires
Interface state:Interface, State

(*, FF07::1), 00:04:45/00:02:47, RP 2001:0DB8:6::6, flags:S
  Incoming interface:Tunnel5
  RPF nbr:6:6:6::6
  Outgoing interface list:
    POS4/0, Forward, 00:04:45/00:02:47

(2001:0DB8:999::99, FF07::1), 00:02:06/00:01:23, flags:SFT
  Incoming interface:POS1/0
  RPF nbr:2001:0DB8:999::99
  Outgoing interface list:
    POS4/0, Forward, 00:02:06/00:03:27

```

Sample Output from the show ipv6 mroute active Command

The following is sample output from the **show ipv6 mroute active** command:

```

Router# show ipv6 mroute active

Active IPv6 Multicast Sources - sending >= 4 kbps
Group:FF05::1

```

```
Source:2001:0DB8:1:1:1
Rate:11 pps/8 kbps(1sec), 8 kbps(last 8 sec)
```

Sample Output from the show ipv6 pim bsr Command

The following example displays BSR election information:

```
Router# show ipv6 pim bsr election
```

```
PIMv2 BSR information
BSR Election Information
Scope Range List: ff00::/8
This system is the Bootstrap Router (BSR)
BSR Address: 2001:0DB8:1:1:4
Uptime: 00:11:55, BSR Priority: 0, Hash mask length: 126
RPF: FE80::A8BB:CCFF:FE03:C400,Ethernet0/0
BS Timer: 00:00:07
This system is candidate BSR
Candidate BSR address: 2001:0DB8:1:1:4, priority: 0, hash mask length: 126
```

Sample Output from the show ipv6 pim group-map Command

The following is sample output from the show ipv6 pim group-map command:

```
Router# show ipv6 pim group-map
```

```
FF33::/32*
SSM
Info source:Static
Uptime:00:08:32, Groups:0
FF34::/32*
SSM
Info source:Static
Uptime:00:09:42, Groups:0
```

Sample Output from the show ipv6 pim interface Command

The following is sample output from the show ipv6 pim interface command using the state-on keyword:

```
Router# show ipv6 pim interface state-on
```

Interface	PIM	Nbr	Hello Count Intvl	DR Prior
Ethernet0	on	0	30	1
Address:FE80::208:20FF:FE08:D7FF				
DR :this system				
POS1/0	on	0	30	1
Address:FE80::208:20FF:FE08:D554				
DR :this system				
POS4/0	on	1	30	1
Address:FE80::208:20FF:FE08:D554				
DR :FE80::250:E2FF:FE8B:4C80				
POS4/1	on	0	30	1
Address:FE80::208:20FF:FE08:D554				
DR :this system				
Loopback0	on	0	30	1
Address:FE80::208:20FF:FE08:D554				
DR :this system				

Sample Output from the show ipv6 pim join-prune statistic Command

The following example provides the join/prune aggregation on Ethernet interface 0/0/0:

```
Router# show ipv6 pim join-prune statistic Ethernet0/0/0
```



```
PIM Average Join/Prune Aggregation for last (1K/10K/50K) packets
Interface                Transmitted                Received

Ethernet0/0/0            0      / 0      / 0            1      / 0      / 0
```

Sample Output from the show ipv6 pim neighbor Command

The following is sample output from the **show ipv6 pim neighbor** command using the **detail** keyword to identify the additional addresses of the neighbors learned through the routable address hello option:

```
Router# show ipv6 pim neighbor detail
```

Neighbor Address(es)	Interface	Uptime	Expires	DR	pri	Bidir
FE80::A8BB:CCFF:FE00:401 60::1:1:3	Ethernet0/0	01:34:16	00:01:16	1		B
FE80::A8BB:CCFF:FE00:501 60::1:1:4	Ethernet0/0	01:34:15	00:01:18	1		B

Sample Output from the show ipv6 pim range-list Command

The following is sample output from the **show ipv6 pim range-list** command:

```
Router# show ipv6 pim range-list
```

```
config SSM Exp:never Learnt from :::
FF33::/32 Up:00:26:33
FF34::/32 Up:00:26:33
FF35::/32 Up:00:26:33
FF36::/32 Up:00:26:33
FF37::/32 Up:00:26:33
FF38::/32 Up:00:26:33
FF39::/32 Up:00:26:33
FF3A::/32 Up:00:26:33
FF3B::/32 Up:00:26:33
FF3C::/32 Up:00:26:33
FF3D::/32 Up:00:26:33
FF3E::/32 Up:00:26:33
FF3F::/32 Up:00:26:33
config SM RP:40::1:1:1 Exp:never Learnt from :::
FF13::/64 Up:00:03:50
config SM RP:40::1:1:3 Exp:never Learnt from :::
FF09::/64 Up:00:03:50
```

Sample Output from the show ipv6 pim topology Command

The following is sample output from the **show ipv6 pim topology** command:

```
Router# show ipv6 pim topology
```

```
IP PIM Multicast Topology Table
Entry state: (*S,G) [RPT/SPT] Protocol Uptime Info
Entry flags:KAT - Keep Alive Timer, AA - Assume Alive, PA - Probe Alive,
RA - Really Alive, LH - Last Hop, DSS - Don't Signal Sources,
RR - Register Received, SR - Sending Registers, E - MSDP External,
DCC - Don't Check Connected
Interface state:Name, Uptime, Fwd, Info
Interface flags:LI - Local Interest, LD - Local Dissinterest,
II - Internal Interest, ID - Internal Dissinterest,
LH - Last Hop, AS - Assert, AB - Admin Boundary

(*,FF05::1)
SM UP:02:26:56 JP:Join(now) Flags:LH
```

```
RP:2001:0DB8:1:1:2
RPF:Ethernet1/1,FE81::1
  Ethernet0/1      02:26:56  fwd LI LH
```

```
(2001:0DB8:1:1:200,FF05::1)
SM UP:00:00:07 JP:Null(never) Flags:
RPF:Ethernet1/1,FE80::30:1:4
  Ethernet1/1      00:00:07  off LI
```

Sample Output from the show ipv6 pim traffic Command

The following example shows the number of PIM protocol messages received and sent.

```
Router# show ipv6 pim traffic
```

```
PIM Traffic Counters
Elapsed time since counters cleared:00:05:29
```

	Received	Sent
Valid PIM Packets	22	22
Hello	22	22
Join-Prune	0	0
Register	0	0
Register Stop	0	0
Assert	0	0
Bidir DF Election	0	0
Errors:		
Malformed Packets		0
Bad Checksums		0
Send Errors		0
Packet Sent on Loopback Errors		0
Packets Received on PIM-disabled Interface		0
Packets Received with Unknown PIM Version		0

Sample Output from the show ipv6 pim tunnel Command

The following is sample output from the **show ipv6 pim tunnel** command on the RP:

```
Router# show ipv6 pim tunnel
```

```
Tunnel0*
Type  :PIM Encap
RP    :100::1
Source:100::1
Tunnel0*
Type  :PIM Decap
RP    :100::1
Source: -
```

The following is sample output from the **show ipv6 pim tunnel** command on a non-RP:

```
Router# show ipv6 pim tunnel
```

```
Tunnel0*
Type  :PIM Encap
RP    :100::1
Source:2001::1:1:1
```

Sample Output from the show ipv6 rpf Command

The following example displays RPF information for the unicast host with the IPv6 address of 2001:0DB8:1:1:2:

```
Router# show ipv6 rpf 2001:0DB8:1:1:2
```

```
RPF information for 2001:0DB8:1:1:2
RPF interface:Ethernet3/2
RPF neighbor:FE80::40:1:3
RPF route/mask:20::/64
RPF type:Unicast
RPF recursion count:0
Metric preference:110
Metric:30
```

Configuration Examples for Implementing IPv6 Multicast

This section provides the following configuration examples:

- [Enabling IPv6 Multicast Routing: Example, page 69](#)
- [Configuring PIM: Examples, page 70](#)
- [Configuring PIM Options: Example, page 70](#)
- [Configuring the MLD Protocol: Examples, page 70](#)
- [Configuring Explicit Tracking of Receivers: Example, page 70](#)
- [Configuring Mroutes: Example, page 71](#)
- [Configuring an IPv6 Multiprotocol BGP Peer Group: Example, page 71](#)
- [Advertising Routes into IPv6 Multiprotocol BGP: Example, page 71](#)
- [Redistributing Prefixes into IPv6 Multiprotocol BGP: Example, page 71](#)
- [Generating Translate Updates for IPv6 Multicast BGP: Example, page 71](#)
- [Disabling Embedded RP Support in IPv6 PIM: Example, page 72](#)
- [Turning Off IPv6 PIM on a Specified Interface: Example, page 72](#)
- [Disabling MLD Router-Side Processing: Example, page 72](#)
- [Disabling and Reenabling MFIB: Example, page 72](#)

Enabling IPv6 Multicast Routing: Example

The following example enables multicast routing on all interfaces. Entering this command also enables multicast forwarding for PIM and MLD on all enabled interfaces of the router.

```
Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
```

Configuring PIM: Examples

The following example shows how to configure a router to use PIM-SM using 2001:0DB8::1 as the RP. The following example sets the SPT threshold to infinity to prevent switchover to the source tree when a source starts sending traffic and sets a filter on all sources that do not have a local multicast BGP prefix.

```
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 pim rp-address 2001:0DB8::1
Router(config)# ipv6 pim spt-threshold infinity
Router(config)# ipv6 pim accept-register route-map reg-filter
```

Configuring PIM Options: Example

The following example sets the DR priority, sets the PIM hello interval, and sets the periodic join and prune announcement interval on Ethernet interface 0/0:

```
Router(config)# interface Ethernet0/0
Router(config)# ipv6 pim hello-interval 60
Router(config)# ipv6 pim dr-priority 3
Router(config)# ipv6 pim join-prune-interval 75
```

Configuring the MLD Protocol: Examples

The following example shows how to configure the query maximum response time, the query timeout, and the query interval on FastEthernet interface 1/0:

```
Router> enable
Router# configure terminal
Router(config)# interface FastEthernet 1/0
Router(config-if)# ipv6 mld query-max-response-time 20
Router(config-if)# ipv6 mld query-timeout 130
Router(config-if)# ipv6 mld query-interval 60
```

The following example configures MLD reporting for a specified group and source, allows the user to perform IPv6 multicast receiver access control, and statically forwards traffic for the multicast group onto FastEthernet interface 1/0:

```
Router> enable
Router# configure terminal
Router(config)# interface FastEthernet 1/0
Router(config)# ipv6 mld join-group FF04::10
Router(config)# ipv6 mld static-group FF04::10 100::1
Router(config)# ipv6 mld access-group acc-grp-1
```

Configuring Explicit Tracking of Receivers: Example

The following example shows how to configure the explicit tracking of receivers:

```
Router> enable
Router# configure terminal
Router(config)# interface FastEthernet 1/0
Router(config-if)# ipv6 mld explicit-tracking list1
```

Configuring Mroutes: Example

The following example shows how to configure a static multicast route to be used for multicast RPF selection only:

```
Router> enable
Router# configure terminal
Router(config)# ipv6 route 2001:0DB8::/64 7::7 100 multicast
```

Configuring an IPv6 Multiprotocol BGP Peer Group: Example

The following example configures the IPv6 multiprotocol BGP peer group named group1:

```
router bgp 65000
no bgp default ipv4-unicast
neighbor group1 peer-group
neighbor 2001:0DB8:0:CC00::1 remote-as 64600

address-family ipv6 multicast
neighbor 3FFE:C00:0:1:A8BB:CCFF:FE00:8200 activate
no auto-summary
no synchronization
exit-address-family
```

Advertising Routes into IPv6 Multiprotocol BGP: Example

The following example injects the IPv6 network 2001:0DB8::/24 into the IPv6 multicast database of the local router. (BGP checks that a route for the network exists in the IPv6 multicast database of the local router before advertising the network.)

```
router bgp 65000
no bgp default ipv4-unicast

address-family ipv6 multicast
network 2001:0DB8::/24
```

Redistributing Prefixes into IPv6 Multiprotocol BGP: Example

The following example redistributes BGP routes into the IPv6 multicast database of the local router:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 multicast
redistribute BGP
```

Generating Translate Updates for IPv6 Multicast BGP: Example

The following example shows how to generate IPv6 multicast BGP updates that correspond to unicast IPv6 updates:

```
router bgp 64900
no bgp default ipv4-unicast
address-family ipv6 multicast
neighbor 2001:0DB8:7000::2 translate-update ipv6 multicast
```

Disabling Embedded RP Support in IPv6 PIM: Example

The following example disables embedded RP support on IPv6 PIM:

```
Router(config)# ipv6 multicast-routing
Router(config)# no ipv6 pim rp embedded
```

Turning Off IPv6 PIM on a Specified Interface: Example

The following example turns off IPv6 PIM on FastEthernet interface 1/0:

```
Router(config)# ipv6 multicast-routing
Router(config)# interface FastEthernet 1/0
Router(config)# no ipv6 pim
```

Disabling MLD Router-Side Processing: Example

The following example turns off MLD router-side processing on FastEthernet interface 1/0:

```
Router> enable
Router# configure terminal
Router(config)# interface FastEthernet 1/0
Router(config-if)# no ipv6 mld router
```

Disabling and Reenabling MFIB: Example

Multicast forwarding is automatically enabled when IPv6 multicast routing is enabled; however, a user may want to disable multicast forwarding on the router. The following example shows how to disable multicast forwarding on the router and, if desired, reenables multicast forwarding on the router. The example also shows how to disable MFIB interrupt-level IPv6 multicast forwarding of outgoing packets on FastEthernet interface 1/0:

```
Router> enable
Router# configure terminal
Router(config)# no ipv6 mfib
Router(config)# ipv6 mfib-mode centralized-only
Router(config)# interface FastEthernet 1/0
Router(config-if)# no ipv6 mfib cef output
```

Additional References

The following sections provide references related to the Implementing Multicast IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 multicast addresses	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide

Related Topic	Document Title
Multicast BGP for IPv6	“Implementing Multiprotocol BGP for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 static routes	“Implementing Static Routes for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 tunnels	“Implementing Tunneling for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IPv4 configuration information	“IP Multicast Features Roadmap,” Cisco IOS IP Multicast Configuration Guide
IPv4 command reference	Cisco IOS IP Multicast Command Reference

Standards

Standard	Title
draft-ietf-pim-sm-v2-new	<i>Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)</i> , March 6, 2003
draft-savola-mboned-mcast-rpaddr	<i>Embedding the Address of RP in IPv6 Multicast Address</i> , May 23, 2003
draft-suz-pim-upstream-detection	<i>PIM Upstream Detection Among Multiple Addresses</i> , February 2003
draft-ietf-pim-bidir-05	<i>Bi-directional Protocol Independent Multicast (BIDIR-PIM)</i> , June 20, 2003
draft-ietf-pim-sm-bsr-03.txt	<i>Bootstrap Router (BSR) Mechanism for PIM Sparse Mode</i> , February 25, 2003

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2373	<i>IP Version 6 Addressing Architecture</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2461	<i>Neighbor Discovery for IP version 6 (IPv6)</i>
RFC 2462	<i>IPv6 Stateless Address Autoconfiguration</i>

RFC	Title
RFC 3576	<i>Change of Authorization</i>
RFC 3590	<i>Source Address Selection for the Multicast Listener Discovery (MLD) Protocol</i>
RFC 3810	<i>Multicast Listener Discovery Version 2 (MLDv2) for IPv6</i>
RFC 4007	<i>IPv6 Scoped Address Architecture</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **aaa authorization multicast default**
- **aaa new-model**
- **clear ipv6 mfib counters**
- **clear ipv6 mld counters**
- **clear ipv6 mld traffic**
- **clear ipv6 multicast aaa authorization**
- **clear ipv6 pim counters**
- **clear ipv6 pim reset**
- **clear ipv6 pim topology**
- **debug ipv6 mfib**
- **debug ipv6 mld**
- **debug ipv6 mld explicit**

- **debug ipv6 mld ssm-map**
- **debug ipv6 mrib client**
- **debug ipv6 mrib io**
- **debug ipv6 mrib proxy**
- **debug ipv6 mrib route**
- **debug ipv6 mrib table**
- **debug ipv6 pim**
- **debug ipv6 pim df-election**
- **ipv6 mfib**
- **ipv6 mfib cef output**
- **ipv6 mfib fast**
- **ipv6 mfib forwarding**
- **ipv6 mfib-mode centralized-only**
- **ipv6 mld access-group**
- **ipv6 mld explicit-tracking**
- **ipv6 mld join-group**
- **ipv6 mld limit**
- **ipv6 mld query-interval**
- **ipv6 mld query-max-response-time**
- **ipv6 mld query-timeout**
- **ipv6 mld router**
- **ipv6 mld snooping**
- **ipv6 mld snooping explicit-tracking**
- **ipv6 mld snooping last-member-query-interval**
- **ipv6 mld snooping limit**
- **ipv6 mld snooping mrouter**
- **ipv6 mld snooping querier**
- **ipv6 mld snooping report-suppression**
- **ipv6 mld ssm-map enable**
- **ipv6 mld ssm-map query dns**
- **ipv6 mld ssm-map static**
- **ipv6 mld state-limit**
- **ipv6 mld static-group**
- **ipv6 multicast aaa account receive**
- **ipv6 multicast boundary scope**
- **ipv6 multicast group-range**
- **ipv6 multicast hardware-switching**
- **ipv6 multicast multipath**

- **ipv6 multicast-routing**
- **ipv6 multicast rpf use-bgp**
- **ipv6 pim**
- **ipv6 pim accept-register**
- **ipv6 pim bsr announced rp**
- **ipv6 pim bsr border**
- **ipv6 pim bsr candidate bsr**
- **ipv6 pim bsr candidate rp**
- **ipv6 pim dr-priority**
- **ipv6 pim hello-interval**
- **ipv6 pim join-prune-interval**
- **ipv6 pim neighbor-filter list**
- **ipv6 pim rp embedded**
- **ipv6 pim rp-address**
- **ipv6 pim spt-threshold infinity**
- **show ipv6 mfib**
- **show ipv6 mfib active**
- **show ipv6 mfib count**
- **show ipv6 mfib interface**
- **show ipv6 mfib status**
- **show ipv6 mfib summary**
- **show ipv6 mld groups**
- **show ipv6 mld groups summary**
- **show ipv6 mld interface**
- **show ipv6 mld snooping**
- **show ipv6 mld ssm-map**
- **show ipv6 mld traffic**
- **show ipv6 mrrib client**
- **show ipv6 mrrib route**
- **show ipv6 mroute**
- **show ipv6 mroute active**
- **show ipv6 pim bsr**
- **show ipv6 pim df**
- **show ipv6 pim df winner**
- **show ipv6 pim group-map**
- **show ipv6 pim interface**
- **show ipv6 pim join-prune statistic**
- **show ipv6 pim neighbor**

- **show ipv6 pim range-list**
- **show ipv6 pim topology**
- **show ipv6 pim traffic**
- **show ipv6 pim tunnel**
- **show ipv6 rpf**

Feature Information for Implementing IPv6 Multicast

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing IPv6 Multicast

Feature Name	Releases	Feature Information
IPv6 Multicast	12.0(26)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(18)SXE 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IPv6 multicast allows a host to send a single data stream to a subset of all hosts simultaneously. This entire document provides information about this feature.
IPv6 multicast: Multicast Listener Discovery (MLD) protocol, versions 1 and 2	12.0(26)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	MLD is used by IPv6 routers to discover multicast listeners (nodes that want to receive multicast packets destined for specific multicast addresses) on directly attached links. There are two versions of MLD: MLD version 1 is based on version 2 of the IGMP for IPv4, and MLD version 2 is based on version 3 of the IGMP for IPv4. IPv6 multicast for Cisco IOS software uses both MLD version 2 and MLD version 1. The following sections provide information about this feature: <ul style="list-style-type: none"> Restrictions for Implementing IPv6 Multicast, page 2

Table 1 **Feature Information for Implementing IPv6 Multicast (continued)**

Feature Name	Releases	Feature Information
		<ul style="list-style-type: none"> • IPv6 Multicast Overview, page 4 • IPv6 Multicast Routing Implementation, page 7 • Multicast Listener Discovery Protocol for IPv6, page 7 • Protocol Independent Multicast, page 9 • MRIB, page 16 • Enabling IPv6 Multicast Routing, page 19 • Configuring the MLD Protocol, page 19 • Configuring SSM Mapping, page 41 • Disabling MLD Router-Side Processing, page 58
IPv6 multicast: PIM sparse mode (PIM-SM)	12.0(26)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>PIM-SM uses unicast routing to provide reverse-path information for multicast tree building. PIM-SM is used in a multicast network when relatively few routers are involved in each multicast and these routers do not forward multicast packets for a group, unless there is an explicit request for the traffic.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing IPv6 Multicast, page 2 • IPv6 Multicast Routing Implementation, page 7 • Protocol Independent Multicast, page 9 • IPv6 Multicast Process Switching and Fast Switching, page 17 • Multiprotocol BGP for the IPv6 Multicast Address Family, page 18 • Enabling IPv6 Multicast Routing, page 19 • Configuring PIM, page 29 • Configuring a BSR and Verifying BSR Information, page 37 • Disabling Embedded RP Support in IPv6 PIM, page 56 • Turning Off IPv6 PIM on a Specified Interface, page 57

Table 1 *Feature Information for Implementing IPv6 Multicast (continued)*

Feature Name	Releases	Feature Information
IPv6 multicast: PIM Source Specific Multicast (PIM-SSM)	12.0(26)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	PIM-SSM supports the implementation of SSM and is derived from PIM-SM. The SSM feature forwards datagram traffic to receivers from only those multicast sources that the receivers have explicitly joined, optimizing bandwidth utilization and denying unwanted Internet broadcast traffic. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Multicast Routing Implementation, page 7 • Protocol Independent Multicast, page 9 • PIM-Source Specific Multicast, page 13 • IPv6 Multicast Process Switching and Fast Switching, page 17 • Configuring PIM, page 29
IPv6 multicast: scope boundaries	12.0(26)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	IPv6 includes support for global and nonglobal addresses. This section describes the usage of IPv6 addresses of different scopes. The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Multicast Addressing, page 4 • Scoped Address Architecture, page 6 • IPv6 BSR, page 12 • Configuring a BSR, page 37
IPv6 multicast: MLD access group	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	The MLD access group provides receiver access control in Cisco IOS IPv6 multicast routers. The following sections provide information about this feature: <ul style="list-style-type: none"> • MLD Access Group, page 9 • Customizing and Verifying MLD on an Interface, page 20
IPv6 multicast: PIM accept register	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	The PIM accept register feature is the ability to perform PIM-SM register message filtering at the RP. The following sections provide information about this feature: <ul style="list-style-type: none"> • PIM-Sparse Mode, page 10 • Configuring PIM Options, page 31

Table 1 **Feature Information for Implementing IPv6 Multicast (continued)**

Feature Name	Releases	Feature Information
IPv6 multicast: PIM embedded RP support	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>Embedded RP support allows the router to learn RP information using the multicast group destination address instead of the statically configured RP.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • PIM-Sparse Mode, page 10 • Disabling Embedded RP Support in IPv6 PIM, page 56
IPv6 multicast: RPF flooding of BSR packets	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The RPF flooding of BSR packets enables a Cisco IOS IPv6 router to not disrupt the flow of BSMs.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 BSR, page 12
IPv6 multicast: routable address hello option	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The routable address hello option adds a PIM hello message option that includes all the addresses on the interface on which the PIM hello message is advertised.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Routable Address Hello Option, page 15 • Configuring PIM Options, page 31
IPv6 multicast: static multicast routing (mroute)	12.0(26)S 12.3(4)T 12.2(25)S 12.2(28)SB 12.2(33)SRA 12.2(33)SXH 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>IPv6 static mroutes share the same database as IPv6 static routes and are implemented by extending static route support.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing IPv6 Multicast, page 2 • Static Mroutes, page 16 • Configuring Static Mroutes, page 42

Table 1 *Feature Information for Implementing IPv6 Multicast (continued)*

Feature Name	Releases	Feature Information
IPv6 multicast: address family support for multiprotocol BGP	12.0(26)S 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(4)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	This feature provides multicast BGP extensions for IPv6 and supports the same features and functionality as IPv4 BGP. The following sections provide information about this feature: <ul style="list-style-type: none"> • Multiprotocol BGP for the IPv6 Multicast Address Family, page 18 • Configuring IPv6 Multiprotocol BGP, page 44
IPv6 multicast: explicit tracking of receivers	12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(7)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	This feature allows a router to track the behavior of the hosts within its IPv6 network. The following sections provide information about this feature: <ul style="list-style-type: none"> • Explicit Tracking of Receivers, page 9 • Configuring Explicit Tracking of Receivers to Track Host Behavior, page 24
IPv6 multicast: IPv6 bidirectional PIM	12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(25)S 12.3(7)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Bidirectional PIM allows multicast routers to keep reduced state information. Bidirectional shared trees convey data from sources to the RP and distribute them from the RP to the receivers. The following sections provide information about this feature: <ul style="list-style-type: none"> • Restrictions for Implementing IPv6 Multicast, page 2 • Bidirectional PIM, page 16 • Configuring Bidirectional PIM and Displaying Bidirectional PIM Information, page 33
IPv6 multicast: MRIB	12.0(26)S 12.2(18)S 12.2(25)SG 12.3(2)T 12.4 12.4(2)T	The MRIB is a protocol-independent repository of multicast routing entries instantiated by multicast routing protocols (routing clients). The following sections provide information about this feature: <ul style="list-style-type: none"> • MRIB, page 16 • Distributed MFIB, page 17 • Clearing the PIM Topology Table to Reset the MRIB Connection, page 35

Table 1 **Feature Information for Implementing IPv6 Multicast (continued)**

Feature Name	Releases	Feature Information
IPv6 multicast: MFIB and MFIB display enhancements	12.0(26)S 12.2(18)S 12.3(2)T 12.4 12.4(2)T	<p>The MFIB is a platform-independent and routing-protocol-independent library for IPv6 software.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing IPv6 Multicast, page 2 • MFIB, page 16 • IPv6 Multicast Process Switching and Fast Switching, page 17 • Using MFIB in IPv6 Multicast, page 54 • Disabling MFIB on the Router, page 58 • Disabling MFIB Interrupt-Level IPv6 Multicast Forwarding, page 60
IPv6 multicast: IPv6 BSR	12.0(28)S 12.2(25)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(33)SXH 12.3(11)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>If an RP becomes unreachable, this feature allows the RP to be detected and the mapping tables modified so that the unreachable RP is no longer used, and the new tables will be rapidly distributed throughout the domain.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 BSR, page 12 • Configuring a BSR, page 37
IPv6 multicast: IPv6 BSR bidirectional support	12.3(14)T 12.4 12.4(2)T	<p>Bidirectional BSR support allows bidirectional RPs to be advertised in C-RP messages and bidirectional ranges in the BSM.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 BSR, page 12
IPv6 multicast: IPv6 BSR scoped-zone support	12.2(18)SXE 12.2(28)SB	<p>BSR provides scoped zone support by distributing group-to-RP mappings in networks using administratively scoped multicast. The user can configure candidate BSRs and a set of candidate RPs for each administratively scoped region in the user's domain.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 BSR, page 12 • Configuring BSR for Use Within Scoped Zones, page 39 • Configuring BSR Routers to Announce Scope-to-RP Mappings, page 40

Table 1 **Feature Information for Implementing IPv6 Multicast (continued)**

Feature Name	Releases	Feature Information
IPv6 multicast: SSM mapping for MLDv1 SSM	12.2(18)SXE 12.4(2)T 12.2(28)SB 12.2(33)SRA Cisco IOS XE Release 2.1	<p>This feature allows deployment of IPv6 SSM with hosts that are incapable of providing MLD version 2 support in their TCP/IP host stack and their IP multicast receiving application.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • SSM Mapping for IPv6, page 14 • Configuring SSM Mapping, page 41
IPv6 multicast: IPv6 BSR—ability to configure RP mapping	12.4(2)T	<p>This feature allows IPv6 multicast routers to be statically configured to announce scope-to-RP mappings directly from the BSR instead of learning them from candidate-RP messages.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 BSR, page 12 • Configuring BSR Routers to Announce Scope-to-RP Mappings, page 40
IPv6 multicast: MLD group limits	12.4(2)T	<p>The MLD group limits feature provides protection against denial of service (DoS) attacks caused by MLD packets.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Multicast Listener Discovery Protocol for IPv6, page 7 • Implementing MLD Group Limits, page 22
IPv6 multicast: multicast user authentication and profile support	12.4(4)T	<p>Multicast access control provides an interface between multicast and AAA for provisioning, authorizing, and accounting at the last-hop router, receiver access control functions in multicast, and group or channel disabling capability in multicast.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Multicast User Authentication and Profile Support, page 9 • Configuring Multicast User Authentication and Profile Support, page 25
IPv6 multicast: process switching and fast switching	12.0(26)S 12.2(18)S 12.3(2)T 12.4 12.4(2)T	<p>In IPv6 multicast process switching, the Route Processor must examine, rewrite, and forward each packet. IPv6 multicast fast switching allows routers to provide better packet forwarding performance than process switching.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 Multicast Process Switching and Fast Switching, page 17

Table 1 **Feature Information for Implementing IPv6 Multicast (continued)**

Feature Name	Releases	Feature Information
Distributed MFIB (dMFIB)	12.0(26)S 12.3(4)T 12.2(25)S 12.4 12.4(2)T 12.2(28)SB	Distributed MFIB dMFIB is used to switch multicast IPv6 packets on distributed platforms. The following sections provide information about this feature: <ul style="list-style-type: none"> • Distributed MFIB, page 17 • Disabling MFIB on a Distributed Platform, page 59
IPv6: Multicast address group range support	12.2(33)SXI	This feature allows the router to keep from receiving multicast traffic to be received from unauthenticated groups or unauthorized channels. The following sections provide information about this feature: <ul style="list-style-type: none"> • Disabling the Router from Receiving Unauthenticated Multicast Traffic, page 27

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003–2009 Cisco Systems, Inc. All rights reserved.



Implementing NAT-PT for IPv6

First Published: November 25, 2002

Last Updated: February 23, 2009

Network Address Translation—Protocol Translation (NAT-PT) is an IPv6-IPv4 translation mechanism, as defined in RFC 2765 and RFC 2766, allowing IPv6-only devices to communicate with IPv4-only devices and vice versa.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing NAT-PT for IPv6” section on page 22](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing NAT-PT for IPv6, page 2](#)
- [Restrictions for Implementing NAT-PT for IPv6, page 2](#)
- [Information About Implementing NAT-PT for IPv6, page 2](#)
- [How to Implement NAT-PT for IPv6, page 5](#)
- [Configuration Examples for NAT-PT for IPv6, page 18](#)
- [Where to Go Next, page 19](#)
- [Additional References, page 19](#)
- [Command Reference, page 21](#)
- [Feature Information for Implementing NAT-PT for IPv6, page 22](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2002–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing NAT-PT for IPv6

Before implementing NAT-PT, you must configure IPv4 and IPv6 on the router interfaces that need to communicate between IPv4-only and IPv6-only networks.

Restrictions for Implementing NAT-PT for IPv6

- NAT-PT is not supported in Cisco Express Forwarding.
- NAT-PT provides limited Application Layer Gateway (ALG) support—ALG support for Internet Control Message Protocol (ICMP), File Transfer Protocol (FTP), and Domain Naming System (DNS).
- NAT-PT has the same restrictions that apply to IPv4 NAT where NAT-PT does not provide end-to-end security and the NAT-PT router can be a single point of failure in the network.
- Users must decide whether to use Static NAT-PT operation, Dynamic NAT-PT operation, Port Address Translation (PAT), or IPv4-mapped operation. Deciding which operation to use determines how a user will configure and operate NAT-PT.

Information About Implementing NAT-PT for IPv6

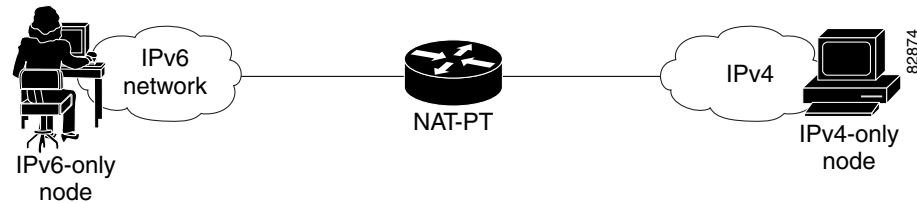
This section provides an overview of NAT-PT for Cisco IOS software. Users can configure NAT-PT using one of the following operations—static NAT-PT, dynamic NAT-PT, Port Address Translation (PAT), or IPv4-mapped operation—which are described in the following sections:

- [NAT-PT Overview, page 2](#)
- [Static NAT-PT Operation, page 3](#)
- [Dynamic NAT-PT Operation, page 4](#)
- [Port Address Translation or Overload, page 5](#)
- [IPv4-Mapped Operation, page 5](#)

NAT-PT Overview

NAT-PT for Cisco IOS software was designed using RFC 2766 and RFC 2765 as a migration tool to help customers transition their IPv4 networks to IPv6 networks. Using a protocol translator between IPv6 and IPv4 allows direct communication between hosts speaking a different network protocol. Users can use either static definitions or IPv4-mapped definitions for NAT-PT operation.

[Figure 30](#) shows NAT-PT runs on a router between an IPv6 network and an IPv4 network to connect an IPv6-only node with an IPv4-only node.

Figure 30 NAT-PT Basic Operation

Although IPv6 solves addressing issues for customers, a long transition period is likely before customers move to an exclusive IPv6 network environment. During the transition period, any new IPv6-only networks will need to continue to communicate with existing IPv4 networks. NAT-PT is designed to be deployed to allow direct communication between IPv6-only networks and IPv4-only networks. For a service provider customer, an example could be an IPv6-only client trying to access an IPv4-only web server. Enterprise customers will also migrate to IPv6 in stages, and many of their IPv4-only networks will be operational for several years. Dual stack networks may have some IPv6-only hosts configured to take advantage of the IPv6 autoconfiguration, global addressing, and simpler management, and these hosts can use NAT-PT to communicate with existing IPv4-only networks in the same organization.

One of the benefits of NAT-PT is that no changes are required to existing hosts, because all the NAT-PT configurations are performed at the NAT-PT router. Customers with existing stable IPv4 networks can introduce an IPv6 network and use NAT-PT to allow communication without disrupting the existing network. To further illustrate the seamless transition, File Transfer Protocol (FTP) can be used between IPv4 and IPv6 networks, just as within an IPv4 network. Packet fragmentation is enabled by default when IPv6 is configured, allowing IPv6 and IPv4 networks to resolve fragmentation problems between the networks. Without the ability to resolve fragmentation, connectivity could become intermittent when fragmented packets might be dropped or improperly interpreted.

Cisco has developed other transition techniques including dual stack, IPv6 over MPLS, and tunneling. NAT-PT should not be used when other native communication techniques exist. If a host is configured as a dual stack host with both IPv4 and IPv6, we do not recommend using NAT-PT to communicate between the dual stack host and an IPv6-only or IPv4-only host. NAT-PT is not recommended for a scenario in which an IPv6-only network is trying to communicate to another IPv6-only network via an IPv4 backbone or vice versa, because NAT-PT would require a double translation to be performed. In this scenario, tunneling techniques would be recommended.

The following sections describe the operations that may be used to configure NAT-PT. Users have the option to use one of the following operations for NAT-PT operation, but not all four.

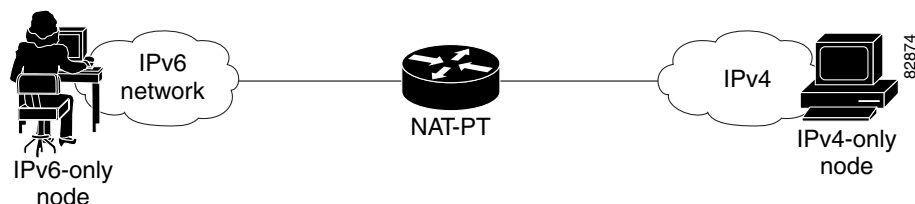
Static NAT-PT Operation

Static NAT-PT uses static translation rules to map one IPv6 address to one IPv4 address. IPv6 network nodes communicate with IPv4 network nodes using an IPv6 mapping of the IPv4 address configured on the NAT-PT router.

[Figure 31](#) shows how the IPv6-only node named A can communicate with the IPv4-only node named C using NAT-PT. The NAT-PT device is configured to map the source IPv6 address for node A of 2001:0db8:bbbb:1::1 to the IPv4 address 192.168.99.2. NAT-PT is also configured to map the source address of IPv4 node C, 192.168.30.1 to 2001:0db8::a. When packets with a source IPv6 address of node A are received at the NAT-PT router, they are translated to have a destination address to match node C in the IPv4-only network. NAT-PT can also be configured to match a source IPv4 address and translate the packet to an IPv6 destination address to allow an IPv4-only host communicate with an IPv6-only host.

If you have multiple IPv6-only or IPv4-only hosts that need to communicate, you may need to configure many static NAT-PT mappings. Static NAT-PT is useful when applications or servers require access to a stable IPv4 address, such as accessing an external IPv4 DNS server.

Figure 31 *Static NAT-PT Operation*

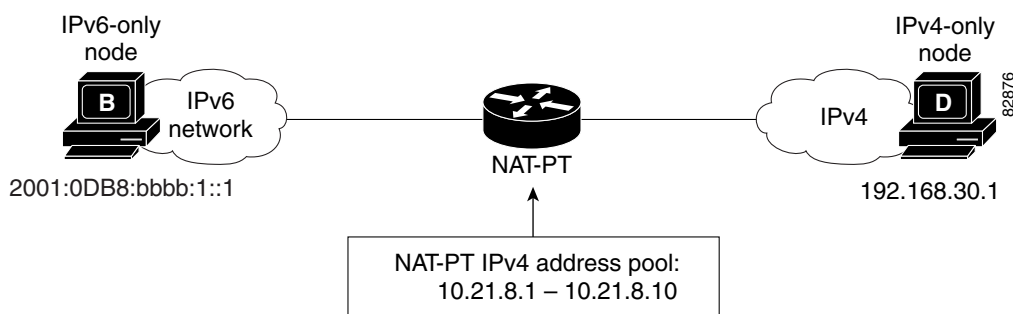


Dynamic NAT-PT Operation

Dynamic NAT-PT allows multiple NAT-PT mappings by allocating addresses from a pool. NAT-PT is configured with a pool of IPv6 and/or IPv4 addresses. At the start of a NAT-PT session a temporary address is dynamically allocated from the pool. The number of addresses available in the address pool determines the maximum number of concurrent sessions. The NAT-PT device records each mapping between addresses in a dynamic state table.

Figure 32 shows how dynamic NAT-PT operates. The IPv6-only node B can communicate with the IPv4-only node D using dynamic NAT-PT. The NAT-PT device is configured with an IPv6 access list, prefix list, or route map to determine which packets are to be translated by NAT-PT. A pool of IPv4 addresses—10.21.8.1 to 10.21.8.10 in Figure 32—is also configured. When an IPv6 packet to be translated is identified, NAT-PT uses the configured mapping rules and assigns a temporary IPv4 address from the configured pool of IPv4 addresses.

Figure 32 *Dynamic NAT-PT Operation*

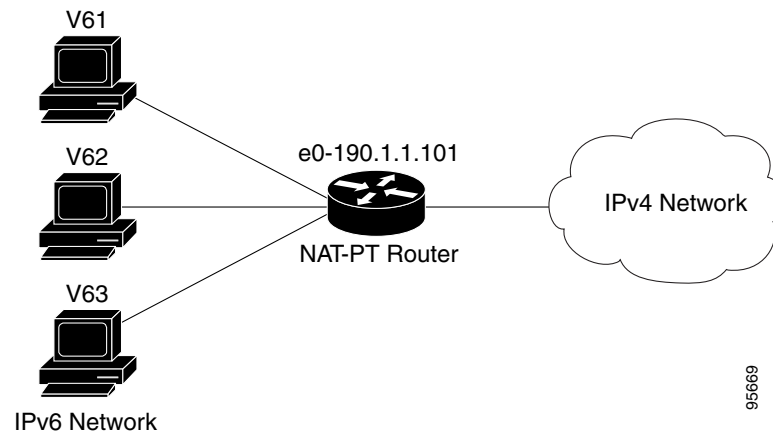


Dynamic NAT-PT translation operation requires at least one static mapping for the IPv4 DNS server. After the IPv6 to IPv4 connection is established, the reply packets going from IPv4 to IPv6 take advantage of the previously established dynamic mapping to translate back from IPv4 to IPv6. If the connection is initiated by an IPv4-only host, then the explanation is reversed.

Port Address Translation or Overload

Port Address Translation (PAT), also known as Overload, allows a single IPv4 address to be used among multiple sessions by multiplexing on the port number to associate several IPv6 users with a single IPv4 address. PAT can be accomplished through a specific interface or through a pool of addresses. [Figure 33](#) shows multiple IPv6 addresses from the IPv6 network linked to a single IPv4 interface into the IPv4 network.

Figure 33 Port Address Translation



IPv4-Mapped Operation

Customers can also send traffic from their IPv6 network to an IPv4 network without configuring IPv6 destination address mapping. A packet arriving at an interface is checked to discover if it has a NAT-PT prefix that was configured with the **ipv6 nat prefix v4-mapped** command. If the prefix matches, then an access-list check is performed to discover if the source address matches the access list or prefix list. If the prefix does not match, the packet is dropped.

If the prefix matches, source address translation is performed. If a rule has been configured for the source address translation, the last 32 bits of the destination IPv6 address is used as the IPv4 destination and a flow entry is created.

How to Implement NAT-PT for IPv6

- [Configuring Basic IPv6 to IPv4 Connectivity for NAT-PT for IPv6, page 6](#) (required)
- [Configuring IPv4-Mapped NAT-PT, page 7](#) (required)
- [Configuring Mappings for IPv6 Hosts Accessing IPv4 Hosts, page 8](#) (required)
- [Configuring Mappings for IPv4 Hosts Accessing IPv6 Hosts, page 11](#) (optional)
- [Configuring PAT for IPv6 to IPv4 Address Mappings, page 12](#)
- [Verifying NAT-PT Configuration and Operation, page 14](#) (optional)

Configuring Basic IPv6 to IPv4 Connectivity for NAT-PT for IPv6

This task explains how to configure basic IPv6 to IPv4 connectivity for NAT-PT, which consists of configuring the NAT-PT prefix globally, and enable NAT-PT on an interface. For NAT-PT to be operational, NAT-PT must be enabled on both the incoming and outgoing interfaces.

NAT-PT Prefix

An IPv6 prefix with a prefix length of 96 must be specified for NAT-PT to use. The IPv6 prefix can be a unique local unicast prefix, a subnet of your allocated IPv6 prefix, or even an extra prefix obtained from your Internet service provider (ISP). The NAT-PT prefix is used to match a destination address of an IPv6 packet. If the match is successful, NAT-PT will use the configured address mapping rules to translate the IPv6 packet to an IPv4 packet. The NAT-PT prefix can be configured globally or with different IPv6 prefixes on individual interfaces. Using a different NAT-PT prefix on several interfaces allows the NAT-PT router to support an IPv6 network with multiple exit points to IPv4 networks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nat prefix *ipv6-prefix/prefix-length***
4. **interface *type number***
5. **ipv6 address *ipv6-prefix* {*/prefix-length* | **link-local**}**
6. **ipv6 nat**
7. **exit**
8. **interface *type number***
9. **ip address *ip-address mask* [secondary]**
10. **ipv6 nat**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 nat prefix <i>ipv6-prefix/prefix-length</i> Example: Router# ipv6 nat prefix 2001:0db8::/96	Assigns an IPv6 prefix as a global NAT-PT prefix. <ul style="list-style-type: none"> Matching destination prefixes in IPv6 packets are translated by NAT-PT. The only prefix length supported is 96.

	Command or Action	Purpose
Step 4	interface <i>type number</i> Example: Router(config)# interface ethernet 3/1	Specifies an interface type and number, and places the router in interface configuration mode.
Step 5	ipv6 address <i>ipv6-address</i> {/prefix-length link-local} Example: Router(config-if)# ipv6 address 2001:0db8:yyyy:1::9/64	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface.
Step 6	ipv6 nat Example: Router(config-if)# ipv6 nat	Enables NAT-PT on the interface.
Step 7	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 8	interface <i>type number</i> Example: Router(config)# interface ethernet 3/3	Specifies an interface type and number, and places the router in interface configuration mode.
Step 9	ip address <i>ip-address mask</i> [secondary] Example: Router(config-if)# ip address 192.168.30.9 255.255.255.0	Specifies an IP address and mask assigned to the interface and enables IP processing on the interface.
Step 10	ipv6 nat Example: Router(config-if)# ipv6 nat	Enables NAT-PT on the interface.

Configuring IPv4-Mapped NAT-PT

The following task describes how to enable customers to send traffic from their IPv6 network to an IPv4 network without configuring IPv6 destination address mapping. This task shows the **ipv6 nat prefix v4-mapped** command configured on a specified interface, but the command could alternatively be configured globally:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 nat prefix** *ipv6-prefix v4-mapped* {*access-list-name* | *ipv6-prefix*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 3/1	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 nat prefix <i>ipv6-prefix</i> v4-mapped <i>{access-list-name ipv6-prefix}</i> Example: Router(config-if)# ipv6 nat prefix 2001::/96 v4-mapped v4mapacl	Enables customers to send traffic from their IPv6 network to an IPv4 network without configuring IPv6 destination address mapping.

Configuring Mappings for IPv6 Hosts Accessing IPv4 Hosts

This task explains how to configure static or dynamic IPv6 to IPv4 address mappings. The dynamic address mappings include assigning a pool of IPv4 addresses and using an access list, prefix list, or route map to define which packets are to be translated.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 nat v6v4 source** *ipv6-address ipv4-address*
or
ipv6 nat v6v4 source *{list access-list-name | route-map map-name}* **pool** *name*
- ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4 prefix-length prefix-length*
- ipv6 nat translation** [**max-entries** *number*] **{timeout | udp-timeout | dns-timeout | tcp-timeout | finrst-timeout | icmp-timeout}** *{seconds | never}*
- ipv6 access-list** *access-list-name*
- permit** *protocol* *{source-ipv6-prefix/prefix-length | any | host source-ipv6-address}* [*operator* *[port-number]*] *{destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address}*
- exit**
- show ipv6 nat translations** [**icmp | tcp | udp**] [**verbose**]
- show ipv6 nat statistics**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 nat v6v4 source <i>ipv6-address</i> <i>ipv4-address</i> or ipv6 nat v6v4 source { list <i>access-list-name</i> route-map <i>map-name</i> } pool <i>name</i> Example: Router(config)# ipv6 nat v6v4 source 2001:0db8:yyyy:1::1 10.21.8.10 or Example: Router(config)# ipv6 nat v6v4 source list pt-list1 pool v4pool	Enables a static IPv6 to IPv4 address mapping using NAT-PT. or Enables a dynamic IPv6 to IPv4 address mapping using NAT-PT. <ul style="list-style-type: none"> Use the list or route-map keyword to specify a prefix list, access list, or a route map to define which packets are translated. Use the pool keyword to specify the name of a pool of addresses, created by the ipv6 nat v6v4 pool command, to be used in dynamic NAT-PT address mapping.
Step 4	ipv6 nat v6v4 pool <i>name</i> <i>start-ipv4</i> <i>end-ipv4</i> prefix-length <i>prefix-length</i> Example: Router(config)# ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24	Specifies a pool of IPv4 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	ipv6 nat translation [max-entries <i>number</i>] { timeout udp-timeout dns-timeout tcp-timeout finrst-timeout icmp-timeout } { <i>seconds</i> never } Example: Router(config)# ipv6 nat translation udp-timeout 600	(Optional) Specifies the time after which NAT-PT translations time out.
Step 6	ipv6 access-list <i>access-list-name</i> Example: Router(config)# ipv6 access-list pt-list1	(Optional) Defines an IPv6 access list and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#. <ul style="list-style-type: none"> The <i>access-list name</i> argument specifies the name of the IPv6 access control list (ACL). IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.

	Command or Action	Purpose
Step 7	<pre>permit protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address}</pre> <p>Example: Router(config-ipv6-acl)# permit ipv6 2001:0db8:bbbb:1::/64 any</p>	<p>(Optional) Specifies permit conditions for an IPv6 ACL.</p> <ul style="list-style-type: none"> The <i>protocol</i> argument specifies the name or number of an Internet protocol. It can be one of the keywords ahp, esp, icmp, ipv6, pcp, sctp, tcp, or udp, or an integer in the range from 0 to 255 representing an IPv6 protocol number. The <i>source-ipv6-prefix/prefix-length</i> and <i>destination-ipv6-prefix/prefix-length</i> arguments specify the source and destination IPv6 network or class of networks about which to set permit conditions. These arguments must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The any keyword is an abbreviation for the IPv6 prefix ::/0. The host source-ipv6-address keyword and argument combination specifies the source IPv6 host address about which to set permit conditions. The <i>source-ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. Only the arguments and keywords relevant to this task are specified here. Refer to the permit command in the <i>IPv6 for Cisco IOS Command Reference</i> document for information on supported arguments and keywords.
Step 8	<pre>exit</pre> <p>Example: Router(config-if)# exit</p>	<p>Exits access list configuration mode, and returns the router to global configuration mode. Enter the exit command twice to return to privileged EXEC mode.</p>
Step 9	<pre>show ipv6 nat translations [icmp tcp udp] [verbose]</pre> <p>Example: Router> show ipv6 nat translations verbose</p>	<p>(Optional) Displays active NAT-PT translations.</p> <ul style="list-style-type: none"> Use the optional icmp, tcp, and udp keywords to display detailed information about the NAT-PT translation events for the specified protocol. Use the optional verbose keyword to display more detailed information about the active translations.
Step 10	<pre>show ipv6 nat statistics</pre> <p>Example: Router> show ipv6 nat statistics</p>	<p>(Optional) Displays NAT-PT statistics.</p>

What to Do Next

If you do not require any IPv4 to IPv6 mappings, proceed to the [“Verifying NAT-PT Configuration and Operation”](#) task.

Configuring Mappings for IPv4 Hosts Accessing IPv6 Hosts

This optional task explains how to configure static or dynamic IPv4 to IPv6 address mappings. The dynamic address mappings include assigning a pool of IPv6 addresses and using an access list, prefix list, or route map to define which packets are to be translated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nat v4v6 source** *ipv4-address* *ipv6-address*
or
ipv6 nat v4v6 source list {*access-list-number* | *name*} **pool** *name*
4. **ipv6 nat v4v6 pool** *name* *start-ipv6* *end-ipv6* **prefix-length** *prefix-length*
5. **access-list** {*access-list-name* | *number*} {**deny** | **permit**} [*source* *source-wildcard*] [**log**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 nat v4v6 source <i>ipv6-address</i> <i>ipv4-address</i>	Enables a static IPv4 to IPv6 address mapping using NAT-PT.
	or	or
	ipv6 nat v4v6 source list { <i>access-list-number</i> <i>name</i> } pool <i>name</i>	Enables a dynamic IPv4 to IPv6 address mapping using NAT-PT.
	Example: Router(config)# ipv6 nat v4v6 source 10.21.8.11 2001:0db8:yyyy::2 or Router(config)# ipv6 nat v4v6 source list 1 pool v6pool	<ul style="list-style-type: none"> • Use the list keyword to specify an access list to define which packets are translated. • Use the pool keyword to specify the name of a pool of addresses, created by the ipv6 nat v4v6 pool command, to be used in dynamic NAT-PT address mapping.

	Command or Action	Purpose
Step 4	ipv6 nat v4v6 pool <i>name start-ipv6 end-ipv6 prefix-length prefix-length</i> Example: Router(config)# ipv6 nat v4v6 pool v6pool 2001:0db8:yyyy::1 2001:0db8:yyyy::2 prefix-length 128	Specifies a pool of IPv6 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	access-list { <i>access-list-name</i> <i>number</i> } { deny permit } [<i>source source-wildcard</i>] [log] Example: Router(config)# access-list 1 permit 192.168.30.0 0.0.0.255	Specifies an entry in a standard IPv4 access list.

Configuring PAT for IPv6 to IPv4 Address Mappings

This task explains how to configure PAT for IPv6 to IPv4 address mappings. Multiple IPv6 addresses are mapped to a single IPv4 address or to a pool of IPv4 addresses and using an access list, prefix list, or route map to define which packets are to be translated.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 nat v6v4 source** {*list access-list-name* | **route-map** *map-name*} **pool** *name overload*
or
ipv6 nat v6v4 source {*list access-list-name* | **route-map** *map-name*} **interface** *interface name overload*
4. **ipv6 nat v6v4 pool** *name start-ipv4 end-ipv4 prefix-length prefix-length*
5. **ipv6 nat translation** [**max-entries** *number*] {**timeout** | **udp-timeout** | **dns-timeout** | **tcp-timeout** | **finrst-timeout** | **icmp-timeout**} {*seconds* | **never**}
6. **ipv6 access-list** *access-list-name*
7. **permit** *protocol* {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 nat v6v4 source {list access-list-name route-map map-name} pool name overload or ipv6 nat v6v4 source {list access-list-name route-map map-name} interface interface name overload Example: Router(config)# ipv6 nat v6v4 source 2001:0db8:yyyy:1::1 10.21.8.10 or Example: Router(config)# ipv6 nat v6v4 source list pt-list1 pool v4pool overload	Enables a dynamic IPv6 to IPv4 address overload mapping using a pool address. or Enables a dynamic IPv6 to IPv4 address overload mapping using an interface address. <ul style="list-style-type: none"> Use the list or route-map keyword to specify a prefix list, access list, or a route map to define which packets are translated. Use the pool keyword to specify the name of a pool of addresses, created by the ipv6 nat v6v4 pool command, to be used in dynamic NAT-PT address mapping. Use the interface keyword to specify the interface address to be used for overload.
Step 4	ipv6 nat v6v4 pool name start-ipv4 end-ipv4 prefix-length prefix-length Example: Router(config)# ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24	Specifies a pool of IPv4 addresses to be used by NAT-PT for dynamic address mapping.
Step 5	ipv6 nat translation [max-entries number] {timeout udp-timeout dns-timeout tcp-timeout finrst-timeout icmp-timeout} {seconds never} Example: Router(config)# ipv6 nat translation udp-timeout 600	(Optional) Specifies the time after which NAT-PT translations time out.

Command or Action	Purpose
<p>Step 6</p> <pre>ipv6 access-list access-list-name</pre> <p>Example:</p> <pre>Router(config)# ipv6 access-list pt-list1</pre>	<p>(Optional) Defines an IPv6 access list and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.</p> <ul style="list-style-type: none"> The <i>access-list name</i> argument specifies the name of the IPv6 access control list (ACL). IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
<p>Step 7</p> <pre>permit protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address}</pre> <p>Example:</p> <pre>Router(config-ipv6-acl)# permit ipv6 2001:0db8:bbbb:1::/64 any</pre>	<p>(Optional) Specifies permit conditions for an IPv6 ACL.</p> <ul style="list-style-type: none"> The <i>protocol</i> argument specifies the name or number of an Internet protocol. It can be one of the keywords ahp, esp, icmp, ipv6, pcp, sctp, tcp, or udp, or an integer in the range from 0 to 255 representing an IPv6 protocol number. The <i>source-ipv6-prefix/prefix-length</i> and <i>destination-ipv6-prefix/prefix-length</i> arguments specify the source and destination IPv6 network or class of networks about which to set permit conditions. These arguments must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. The any keyword is an abbreviation for the IPv6 prefix <code>::/0</code>. The host <i>source-ipv6-address</i> keyword and argument combination specifies the source IPv6 host address about which to set permit conditions. The <i>source-ipv6-address</i> argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons. Only the arguments and keywords relevant to this task are specified here. Refer to the permit command in the Cisco IOS IPv6 Command Reference for information on supported arguments and keywords.

What to Do Next

If you do not require any IPv6-to-IPv4 or IPv4-to-IPv6 mappings, proceed to the [“Verifying NAT-PT Configuration and Operation”](#) task.

Verifying NAT-PT Configuration and Operation

This task explains how to display information to verify the configuration and operation of NAT-PT.

SUMMARY STEPS

1. `clear ipv6 nat translation *`
2. `enable`
3. `debug ipv6 nat [detailed | port]`

DETAILED STEPS

	Command or Action	Purpose
Step 1	clear ipv6 nat translation * Example: Router> <code>clear ipv6 nat translation *</code>	(Optional) Clears dynamic NAT-PT translations from the dynamic translation state table. <ul style="list-style-type: none"> Use the * keyword to clear all dynamic NAT-PT translations. Note Static translation configuration is not affected by this command.
Step 2	enable Example: Router> <code>enable</code>	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 3	debug ipv6 nat [detailed port] Example: Router# <code>debug ipv6 nat detail</code>	Displays debugging messages for NAT-PT translation events.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 nat translations Command](#)
- [Sample Output from the show ipv6 nat statistics Command](#)
- [Sample Output from the clear ipv6 nat translation Command](#)
- [Sample Output from the debug ipv6 nat Command](#)

Sample Output from the show ipv6 nat translations Command

In the following example, output information about active NAT-PT translations is displayed using the `show ipv6 nat translations` command:

Router# `show ipv6 nat translations`

```

Prot  IPv4 source          IPv6 source
     IPv4 destination    IPv6 destination
---  ---                ---
      192.168.123.2      2001:0db8::2

---  ---                ---
      192.168.122.10     2001:0db8::10

tcp   192.168.124.8,11047    2001:0db8:3::8,11047
      192.168.123.2,23   2001:0db8::2,23

```

```

udp  192.168.124.8,52922      2001:0db8:3::8,52922
      192.168.123.2,69        2001::2,69

udp  192.168.124.8,52922      2001:0db8:3::8,52922
      192.168.123.2,52922     2001:0db8::2,52922

---  192.168.124.8            2001:0db8:3::8
      192.168.123.2          2001:0db8::2

---  192.168.124.8            2001:0db8:3::8
      ---                    ---

---  192.168.121.4            2001:0db8:5::4
      ---                    ---

```

In the following example, detailed output information about active NAT-PT translations is displayed using the **show ipv6 nat translations** command with the **verbose** keyword:

```
Router# show ipv6 nat translations verbose
```

```

Prot  IPv4 source      IPv6 source
      IPv4 destination  IPv6 destination
---  ---
      192.168.123.2      2001:0db8::2
      create 00:04:24, use 00:03:24,

---  ---
      192.168.122.10     2001:0db8::10
      create 00:04:24, use 00:04:24,

tcp   192.168.124.8,11047    2001:0db8:3::8,11047
      192.168.123.2,23     2001:0db8::2,23
      create 00:03:24, use 00:03:20, left 00:16:39,

udp   192.168.124.8,52922    2001:0db8:3::8,52922
      192.168.123.2,69     2001:0db8::2,69
      create 00:02:51, use 00:02:37, left 00:17:22,

udp   192.168.124.8,52922    2001:0db8:3::8,52922
      192.168.123.2,52922  2001:0db8::2,52922
      create 00:02:48, use 00:02:30, left 00:17:29,

---  192.168.124.8            2001:0db8:3::8
      192.168.123.2          2001:0db8::2
      create 00:03:24, use 00:02:34, left 00:17:25,

---  192.168.124.8            2001:0db8:3::8
      ---                    ---
      create 00:04:24, use 00:03:24,

---  192.168.121.4            2001:0db8:5::4
      ---                    ---
      create 00:04:25, use 00:04:25,

```

Sample Output from the show ipv6 nat statistics Command

In the following example, output information about NAT-PT statistics is displayed using the **show ipv6 nat statistics** command:

```
Router# show ipv6 nat statistics
```

```

Total active translations: 4 (4 static, 0 dynamic; 0 extended)
NAT-PT interfaces:
  Ethernet3/1, Ethernet3/3

```

```
Hits: 0 Misses: 0
Expired translations: 0
```

Sample Output from the clear ipv6 nat translation Command

In the following example, all dynamic NAT-PT translations are cleared from the dynamic translation state table using the **clear ipv6 nat translation** command with the ***** keyword. When the output information about active NAT-PT translations is then displayed using the **show ipv6 nat translations** command, only the static translation configurations remain. Compare this **show** command output with the output from the **show ipv6 nat translations** command in Step 1.

```
Router# clear ipv6 nat translation *
```

```
Router# show ipv6 nat translations
```

Prot	IPv4 source	IPv6 source
	IPv4 destination	IPv6 destination
---	---	---
	192.168.123.2	2001:0db8::2
---	---	---
	192.168.122.10	2001:0db8::10
---	192.168.124.8	2001:0db8:3::8
---	---	---
---	192.168.121.4	2001:0db8:5::4
---	---	---

Sample Output from the debug ipv6 nat Command

In the following example, debugging messages for NAT-PT translation events are displayed using the **debug ipv6 nat** command:

```
Router# debug ipv6 nat
```

```
00:06:06: IPv6 NAT: icmp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: icmp src (192.168.123.2) -> (2001:0db8:2001::2), dst (192.168.124.8)
-> (2001:0db8:3002::8)
00:06:06: IPv6 NAT: icmp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: icmp src (192.168.123.2) -> (2001:0db8:2001::2), dst (192.168.124.8)
-> (2001:0db8:3002::8)
00:06:06: IPv6 NAT: tcp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: tcp src (192.168.123.2) -> (2001:0db8:2001::2), dst (192.168.124.8) ->
(2001:0db8:3002::8)
00:06:06: IPv6 NAT: tcp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: tcp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: tcp src (2001:0db8:3002::8) -> (192.168.124.8), dst
(2001:0db8:2001::2) -> (192.168.123.2)
00:06:06: IPv6 NAT: tcp src (192.168.123.2) -> (2001:0db8:2001::2), dst (192.168.124.8) ->
(2001:0db8:3002::8)
```

Configuration Examples for NAT-PT for IPv6

This section provides the following configuration examples:

- [Static NAT-PT Configuration: Example, page 18](#)
- [Enabling Traffic to be Sent from an IPv6 Network to an IPv4 Network without Using IPv6 Destination Address Mapping: Example, page 18](#)
- [Dynamic NAT-PT Configuration for IPv6 Hosts Accessing IPv4 Hosts: Example, page 18](#)
- [Dynamic NAT-PT Configuration for IPv4 Hosts Accessing IPv6 Hosts Example, page 19](#)

Static NAT-PT Configuration: Example

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures two static NAT-PT mappings. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```
interface Ethernet3/1
  ipv6 address 2001:0db8:3002::9/64
  ipv6 enable
  ipv6 nat
!
interface Ethernet3/3
  ip address 192.168.30.9 255.255.255.0
  ipv6 nat
!
ipv6 nat v4v6 source 192.168.30.1 2001:0db8:0::2
ipv6 nat v6v4 source 2001:0db8:bbbb:1::1 10.21.8.10
ipv6 nat prefix 2001:0db8:0::/96
```

Enabling Traffic to be Sent from an IPv6 Network to an IPv4 Network without Using IPv6 Destination Address Mapping: Example

In the following example, the access list permits any IPv6 source address with the prefix 2001::/96 to go to the destination with a 2000::/96 prefix. The destination is then translated to the last 32 bit of its IPv6 address; for example: source address = 2001::1, destination address = 2000::192.168.1.1. The destination then becomes 192.168.1.1 in the IPv4 network:

```
ipv6 nat prefix 2000::/96 v4-mapped v4map_acl

ipv6 access-list v4map_acl
  permit ipv6 2001::/96 2000::/96
```

Dynamic NAT-PT Configuration for IPv6 Hosts Accessing IPv4 Hosts: Example

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures one static NAT-PT mapping (used, for example, to access a DNS server). A dynamic NAT-PT mapping is also configured to map IPv6 addresses to IPv4 addresses using a pool of IPv4 addresses named v4pool. The packets to be translated by NAT-PT are filtered using an IPv6 access list named pt-list1. The User Datagram Protocol (UDP) translation entries are configured to time out after 10 minutes. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```

interface Ethernet3/1
  ipv6 address 2001:0db8:bbbb:1::9/64
  ipv6 enable
  ipv6 nat
!
interface Ethernet3/3
  ip address 192.168.30.9 255.255.255.0
  ipv6 nat
!
ipv6 nat v4v6 source 192.168.30.1 2001:0db8:0::2
ipv6 nat v6v4 source list pt-list1 pool v4pool
ipv6 nat v6v4 pool v4pool 10.21.8.1 10.21.8.10 prefix-length 24
ipv6 nat translation udp-timeout 600
ipv6 nat prefix 2001:0db8:1::/96
!
ipv6 access-list pt-list1
  permit ipv6 2001:0db8:bbbb:1::/64 any

```

Dynamic NAT-PT Configuration for IPv4 Hosts Accessing IPv6 Hosts Example

The following example configures the NAT-PT prefix globally, enables NAT-PT on two interfaces, and configures one static NAT-PT mapping (used, for example, to access a DNS server). A dynamic NAT-PT mapping is also configured to map IPv4 addresses to IPv6 addresses using a pool of IPv6 addresses named v6pool. The packets to be translated by NAT-PT are filtered using an access list named pt-list2. Ethernet interface 3/1 is configured as IPv6 only, and Ethernet interface 3/3 is configured as IPv4 only.

```

interface Ethernet3/1
  ipv6 address 2001:0db8:bbbb:1::9/64
  ipv6 enable
  ipv6 nat
!
interface Ethernet3/3
  ip address 192.168.30.9 255.255.255.0
  ipv6 nat
!
ipv6 nat v4v6 source list pt-list2 pool v6pool
ipv6 nat v4v6 pool v6pool 2001:0db8:0::1 2001:0db8:0::2 prefix-length 128
ipv6 nat v6v4 source 2001:0db8:bbbb:1::1 10.21.8.0
ipv6 nat prefix 2001:0db8:0::/96
!
access-list pt-list2 permit 192.168.30.0 0.0.0.255

```

Where to Go Next

If you want to implement IPv6 routing protocols, refer to the [Implementing RIP for IPv6](#), [Implementing IS-IS for IPv6](#), or the [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing NAT-PT for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IP addressing and IP addressing services	Implementing IPv6 Addressing and Basic Connectivity
IP addressing and IP addressing services commands	Cisco IOS IP Addressing Services Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2765	<i>Stateless IP/ICMP Translation Algorithm (SIIT)</i>
RFC 2766	<i>Network Address Translation - Protocol Translation (NAT-PT)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **clear ipv6 nat translation**
- **debug ipv6 nat**
- **ipv6 nat**
- **ipv6 nat max-entries**
- **ipv6 nat prefix**
- **ipv6 nat prefix v4-mapped**
- **ipv6 nat translation**
- **ipv6 nat v4v6 pool**
- **ipv6 nat v4v6 source**
- **ipv6 nat v6v4 pool**
- **ipv6 nat v6v4 source**
- **show ipv6 nat statistics**
- **show ipv6 nat translations**

Feature Information for Implementing NAT-PT for IPv6

Table 16 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(13)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#).

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 16 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 16 Feature Information for Implementing NAT-PT for IPv6

Feature Name	Releases	Feature Information
NAT Protocol Translation	12.2(13)T 12.3 12.3(2)T 12.4	NAT-PT is an IPv6-IPv4 translation mechanism that allows IPv6-only devices to communicate with IPv4-only devices and vice versa. NAT-PT is not supported in Cisco Express Forwarding. This entire document provides information about this feature.
NAT-PT: support for DNS ALG	12.2(13)T 12.3 12.3(2)T 12.4	IPv6 provides DNS ALG support. The following section provides information about this feature: <ul style="list-style-type: none"> Restrictions for Implementing NAT-PT for IPv6, page 2
NAT-PT: support for overload (PAT)	12.3(2)T 12.4 12.4(2)T	PAT, also known as Overload, allows a single IPv4 address to be used among multiple sessions by multiplexing on the port number to associate several IPv6 users with a single IPv4 address. The following sections provide information about this feature: <ul style="list-style-type: none"> Port Address Translation or Overload, page 5 Configuring PAT for IPv6 to IPv4 Address Mappings, page 12 Verifying NAT-PT Configuration and Operation, page 14

Table 16 **Feature Information for Implementing NAT-PT for IPv6 (continued)**

Feature Name	Releases	Feature Information
NAT-PT: support for FTP ALG	12.3(2)T 12.4 12.4(2)T	IPv6 provides FTP ALG support. The following section provides information about this feature: <ul style="list-style-type: none"> • Restrictions for Implementing NAT-PT for IPv6, page 2
NAT-PT: support for fragmentation	12.3(2)T 12.4 12.4(2)T	Packet fragmentation is enabled by default when IPv6 is configured, allowing IPv6 and IPv4 networks to resolve fragmentation problems between the networks. The following section provides information about this feature: <ul style="list-style-type: none"> • NAT-PT Overview, page 2

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing NetFlow for IPv6

First Published: June 26, 2006
Last Updated: July 11, 2008



Note

Effective with Cisco IOS Release 12.4(20)T, the NetFlow for IPv6 feature has been replaced by the IPv6 Flexible NetFlow feature. For information on this feature, see the [Cisco IOS Flexible NetFlow Features Roadmap](#).

NetFlow for IPv6 provides basic NetFlow functionality for IPv6 without affecting IPv4 NetFlow performance.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing NetFlow for IPv6” section on page 14](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Prerequisites for Implementing NetFlow for IPv6, page 2](#)
- [Information About Implementing NetFlow for IPv6, page 2](#)
- [How to Implement NetFlow for IPv6, page 2](#)
- [Configuration Examples for Implementing NetFlow for IPv6, page 11](#)
- [Additional References, page 12](#)
- [Command Reference, page 13](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006–2008 Cisco Systems, Inc. All rights reserved.

- [Feature Information for Implementing NetFlow for IPv6, page 14](#)

Prerequisites for Implementing NetFlow for IPv6

This document assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Additional References](#)” section for IPv4 configuration and command reference information.

Information About Implementing NetFlow for IPv6

To configure NetFlow for IPv6 for Cisco IOS software, you should understand the following concept:

- [NetFlow for IPv6 Environments, page 2](#)

NetFlow for IPv6 Environments

NetFlow for IPv6 is based on NetFlow Version 9 and functions by identifying packet flows for ingress IP and IPv6 packets. NetFlow enables you to collect traffic flow statistics on your routing devices and analyze traffic patterns, such as tasks that are used to perform traffic analysis and detect denial of service (DoS) attacks. It does not involve any connection-setup protocol between routers or to any other networking device or end station and does not require any change externally—either to the traffic or packets themselves or to any other networking device.

NetFlow is completely transparent to the existing network, including end stations and application software and network devices such as LAN switches. Also, NetFlow is performed independently on each internetworking device; it need not be operational on each router in the network. You can use NetFlow Data Export (NDE) to export data to a remote workstation for data collection and further processing. Network planners can selectively invoke NDE on a router or on a per-subinterface basis to gain traffic performance, control, or accounting benefits in specific network locations. NetFlow collects accounting information for IPv6 encapsulation and tunnels. If NetFlow capture is configured on a logical interface, IPv6 flows will be reported with that interface as the input or output interface, depending on whether the feature has been activated on the ingress or egress port.

How to Implement NetFlow for IPv6

To configure NetFlow for IPv6, you must define the exporting scheme that will be used to export NetFlow statistics, configure the NetFlow cache, and configure NetFlow on the interfaces from which statistics will be gathered. The tasks required to complete perform these functions are described in the following sections:

- [Defining the Exporting Scheme Used to Gather NetFlow for IPv6 Statistics, page 3](#) (required)
- [Customizing the NetFlow for IPv6 Cache, page 4](#) (optional)
- [Managing NetFlow for IPv6 Statistics, page 6](#) (optional)
- [Configuring an Aggregation Cache for NetFlow for IPv6, page 6](#) (optional)
- [Configuring a NetFlow for IPv6 Minimum Prefix Mask for Router-Based Aggregation, page 8](#) (optional)

Defining the Exporting Scheme Used to Gather NetFlow for IPv6 Statistics

This task describes how to define the exporting scheme that is used to gather NetFlow for IPv6 statistics.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flow-export version 9** [*origin-as* | *peer-as*] [*bgp-nexthop*]
4. **ipv6 flow-export destination** *ip-address* *udp-port*
5. **ipv6 flow-export template** {*refresh-rate* *packet-refresh-rate* | **timeout** *timeout-value*}
6. **ipv6 flow-export template options** {**export-stats** | *refresh-rate* *packet-refresh-rate* | **timeout** *timeout-value*}
7. **interface** *type number*
8. **ipv6 flow** {*ingress* | *egress*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 flow-export version 9 [<i>origin-as</i> <i>peer-as</i>] [<i>bgp-nexthop</i>] Example: Router(config)# ipv6 flow-export version 9	Enables NetFlow routing.
Step 4	ipv6 flow-export destination <i>ip-address</i> <i>udp-port</i> Example: Router(config)# ipv6 flow-export destination 10.0.101.254 9991	Enables the exporting of information in NetFlow cache entries to a specific address or port.
Step 5	ipv6 flow-export template { <i>refresh-rate</i> <i>packet-refresh-rate</i> timeout <i>timeout-value</i> } Example: Router(config)# ipv6 flow-export template timeout 60	Enables the exporting of information in NetFlow cache entries.

	Command or Action	Purpose
Step 6	ipv6 flow-export template options <code>{export-stats refresh-rate packet-refresh-rate timeout timeout-value}</code> Example: Router(config)# ipv6 flow-export template options export-stats	Configures templates for IPv6 cache exports.
Step 7	interface type number Example: Router(config)# interface atm 0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	ipv6 flow {ingress egress} Example: Router(config-if)# ipv6 flow ingress	(Optional) Enables IPv6 flow capture for incoming (ingress) or outgoing (egress) packets. Commands for ingress and egress can be specified on the same interface. If a switched packet belongs to a flow that is captured at both ingress and egress, it will be counted twice. This command must be entered on each interface and for each direction in which NetFlow capture is needed.

Customizing the NetFlow for IPv6 Cache

Several options are available for configuring and customizing the NetFlow for IPv6 cache:

- Customize the number of entries in the NetFlow for IPv6 cache
- Customize the timeout
- Customize the Multiprotocol Label Switching (MPLS) parameters

These options are described in the following optional task:

- [Customizing the NetFlow for IPv6 Cache, page 4](#)

Customizing the NetFlow for IPv6 Cache

Normally, the size of the NetFlow for IPv6 cache will meet your needs. However, you can increase or decrease the number of entries maintained in the cache to meet the needs of your NetFlow traffic rates. The default is 64K flow cache entries. Each cache entry requires about 64 bytes of storage. Assuming a cache with the default number of entries, about 4 MB of DRAM would be required. Each time a new flow is taken from the free flow queue, the number of free flows is checked. If only a few free flows remain, NetFlow attempts to age 30 flows using an accelerated timeout. If only 1 free flow remains, NetFlow automatically ages 30 flows regardless of their age. The intent is to ensure that free flow entries are always available.



Caution

Cisco recommends that you not change the number of NetFlow cache entries. Improper use of this feature could cause network problems. To return to the default NetFlow cache entries, use the **no ip flow-cache entries** global configuration command.

The following task describes how to customize the number of entries in the NetFlow cache.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flow-cache entries** *number*
4. **ipv6 flow-cache timeout** {**active** *minutes* | **inactive** *seconds*}
5. **ipv6 flow-aggregation cache** {**as** | **bgp-nexthop** | **destination-prefix** | **prefix** | **protocol-port** | **source-prefix**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 flow-cache entries <i>number</i> Example: Router(config)# ipv6 flow-cache entries 131072	Changes the number of entries maintained in the NetFlow cache.
Step 4	ipv6 flow-cache timeout { active <i>minutes</i> inactive <i>seconds</i> } Example: Router(config)# ipv6 flow-cache timeout active 10	Changes the timeout values for the NetFlow cache.
Step 5	ipv6 flow-aggregation cache { as bgp-nexthop destination-prefix prefix protocol-port source-prefix } Example: Router(config)# ipv6 flow-aggregation cache as	Configures the aggregation cache configuration scheme.

Managing NetFlow for IPv6 Statistics

You can display and clear NetFlow for IPv6 statistics. NetFlow for IPv6 statistics consist of IPv6 packet size distribution, IP flow cache information, and flow information such as the protocol, total flow, and flows per second. The resulting information can be used to determine information about your router traffic.

The following task describes how to manage NetFlow for IPv6 statistics. Use these commands as needed for verification of configuration.

SUMMARY STEPS

1. **enable**
2. **show ip cache flow**
3. **clear ip flow stats**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	show ip cache flow Example: Router# show ip cache flow	Displays NetFlow statistics.
Step 3	clear ip flow stats Example: Router# clear ip flow stats	Clears the NetFlow statistics.

Configuring an Aggregation Cache for NetFlow for IPv6

The following task describes how to configure an aggregation cache for NetFlow for IPv6.

Prerequisites

To configure an aggregation cache, you must enter aggregation cache configuration mode, and you must decide which type of aggregation scheme you want to configure: Autonomous System, Destination Prefix, Prefix, Protocol Prefix, or Source Prefix aggregation cache. Once you define the aggregation scheme, the following task lets you define the operational parameters for that scheme.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **ipv6 flow-export destination** *ip-address udp-port*
4. **ipv6 flow-aggregation cache** {*as* | **bgp-nexthop** | **destination-prefix** | **prefix** | **protocol-port** | **source-prefix**}
5. **cache** {*entries number* | **timeout** {*active minutes* | *inactive seconds*}}
6. **cache** {*entries number* | **timeout** {*active minutes* | *inactive seconds*}}
7. **exit**
8. **ipv6 flow-export destination** *ip-address udp-port*

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 flow-export destination <i>ip-address udp-port</i> Example: Router(config)# ipv6 flow-export destination 10.42.42.1 9991	Enables the exporting of information in NetFlow cache entries to a specific address or port.
Step 4	ipv6 flow-aggregation cache { <i>as</i> bgp-nexthop destination-prefix prefix protocol-port source-prefix } Example: Router(config)# ipv6 flow-aggregation cache as	Configures the aggregation cache configuration scheme, and places the router in NetFlow aggregation cache configuration mode.
Step 5	cache { <i>entries number</i> timeout { <i>active minutes</i> <i>inactive seconds</i> }} Example: Router(config-flow-cache)# cache entries 2046	Specifies the number (in this example, 2046) of cache entries to allocate for the autonomous system aggregation cache.
Step 6	cache { <i>entries number</i> timeout { <i>active minutes</i> <i>inactive seconds</i> }} Example: Router(config-flow-cache)# cache timeout inactive 199	Specifies the number of seconds (in this example, 199) that an inactive entry is allowed to remain in the aggregation cache before it is deleted.

	Command	Purpose
Step 7	exit Example: Router(config-flow-cache)# exit	Exits NetFlow aggregation cache configuration mode, and places the router in global configuration mode.
Step 8	ipv6 flow-export destination <i>ip-address</i> <i>udp-port</i> Example: Router(config)# ipv6 flow-export destination 10.0.101.254 9991	Enables the data export.

Configuring a NetFlow for IPv6 Minimum Prefix Mask for Router-Based Aggregation

To configure the NetFlow for IPv6 Minimum Prefix Mask for Router-Based Aggregation feature, perform the tasks described in the following sections. Each task is optional.

- [Configuring the Minimum Mask of a Prefix Aggregation Scheme, page 8](#)
- [Configuring the Minimum Mask of a Destination-Prefix Aggregation Scheme, page 9](#)
- [Configuring the Minimum Mask of a Source-Prefix Aggregation Scheme, page 10](#)

Configuring the Minimum Mask of a Prefix Aggregation Scheme

The following task describes how to configure the minimum mask of a prefix aggregation scheme.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flow-aggregation cache** {as | bgp-nexthop | destination-prefix | prefix | protocol-port | source-prefix}
4. **mask** {destination | source} **minimum** *value*

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 3	ipv6 flow-aggregation cache {as bgp-nexthop destination-prefix prefix protocol-port source-prefix } Example: Router(config)# ipv6 flow-aggregation cache prefix	Configures the aggregation cache configuration scheme, and places the router in NetFlow aggregation cache configuration mode.
Step 4	mask { destination source } minimum value Example: Router(config-flow-cache)# mask source minimum value	Configures the minimum value for the source mask.

Configuring the Minimum Mask of a Destination-Prefix Aggregation Scheme

The following task describes how to configure the minimum mask of a destination-prefix aggregation scheme.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flow-aggregation cache** {as | **bgp-nexthop** | **destination-prefix** | **prefix** | **protocol-port** | **source-prefix**}
4. **mask** {**destination** | **source**} **minimum value**

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 3	ipv6 flow-aggregation cache {as bgp-nexthop destination-prefix prefix protocol-port source-prefix} Example: Router(config)# ipv6 flow-aggregation cache destination-prefix	Configures the aggregation cache configuration scheme, and places the router in NetFlow aggregation cache configuration mode.
Step 4	mask {destination source} minimum <i>value</i> Example: Router(config-flow-cache)# mask destination minimum 32	Configures the minimum value for the destination mask.

Configuring the Minimum Mask of a Source-Prefix Aggregation Scheme

The following task describes how to configure the minimum mask of a source-prefix aggregation scheme.



Note

If the minimum mask has not been explicitly configured, no minimum mask information is displayed. The default value of the minimum mask is zero. The configurable range for the minimum mask is from 1 to 32. An appropriate value should be chosen by the user depending on the traffic. A higher value of the minimum mask will provide more detailed network addresses, but it may also result in an increased number of flows in the aggregation cache.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 flow-aggregation cache** {as | bgp-nexthop | destination-prefix | prefix | protocol-port | source-prefix}
4. **mask** {destination | source} **minimum** *value*

DETAILED STEPS

	Command	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command	Purpose
Step 3	ipv6 flow-aggregation cache {as bgp-nexthop destination-prefix prefix protocol-port source-prefix} Example: Router(config)# ipv6 flow-aggregation cache source-prefix	Configures the aggregation cache configuration scheme, and places the router in NetFlow aggregation cache configuration mode.
Step 4	mask {destination source} minimum value Example: Router(config-flow-cache)# mask source minimum 5	Configures the minimum value for the source mask.

Configuration Examples for Implementing NetFlow for IPv6

The section provides the following configuration example:

- [Configuring NetFlow in IPv6 Environments: Example, page 11](#)

Configuring NetFlow in IPv6 Environments: Example

If you configure the **ipv6 flow ingress** command on a few selected subinterfaces and then configure the **ip route-cache flow** command on the main interface, enabling the main interface will overwrite the **ip flow ingress** command and data collection will start from the main interface and from all the subinterfaces. In a scenario where you configure the **ipv6 flow ingress** command and then configure the **ip route-cache flow** command on the main interface, you can restore subinterface data collection by using the **no ip route-cache flow** command. This configuration will disable data collection from the main interface and restore data collection to the subinterfaces you originally configured with the **ipv6 flow ingress** command.

The following example shows how to configure NetFlow on Fast Ethernet subinterface 6/3.0:

```
Router(config)# interface FastEthernet6/3.0
Router(config-subif)# ipv6 flow ingress
```

The following example shows the configuration for a loopback source interface. The loopback interface has the IPv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64 and is used by the serial interface in slot 5, port 0.

```
Router# configure terminal
Router(config)# interface loopback 0
Router(config-if)# ipv6 address 2001:0DB8:1:1:FFFF:FFFF:FFFF:FFFE/64
Router(config-if)# exit
Router(config)# interface serial 5/0:0
Router(config-if)# ip unnumbered loopback0
Router(config-if)# encapsulation ppp
Router(config-if)# ipv6 flow cache
Router(config-if)# exit
Router(config)# ipv6 flow-export source loopback 0
Router(config)# exit
```

Additional References

The following sections provide references related to the Implementing NetFlow for IPv6 feature.

Related Documents

Related Topic	Document Title
Cisco IOS Flexible NetFlow	Cisco IOS Flexible NetFlow Features Roadmap
NetFlow for IPv4 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS NetFlow Command Reference
NetFlow for IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 supported feature list	Start Here: Cisco IOS Software Release Specifics for IPv6 Features

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **ipv6 flow**
- **ipv6 flow ingress**
- **ipv6 flow-aggregation cache**
- **ipv6 flow-cache entries**
- **ipv6 flow-cache timeout**
- **ipv6 flow-export destination**
- **ipv6 flow-export source**
- **ipv6 flow-export template**
- **ipv6 flow-export template options**
- **ipv6 flow-export version 9**

Feature Information for Implementing NetFlow for IPv6

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing NetFlow for IPv6

Feature Name	Releases	Feature Information
IPv6: NetFlow for IPv6	12.3(7)T 12.4 12.4(2)T	NetFlow for IPv6 enables you to collect traffic flow statistics on your routing devices and analyze traffic patterns, which are used to detect DoS attacks. The following sections provide information about this feature: <ul style="list-style-type: none"> NetFlow for IPv6 Environments, page 2 How to Implement NetFlow for IPv6, page 2
NetFlow: Removal of IPv6 NetFlow	12.4(20)T	This feature was removed.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2006–2009 Cisco Systems, Inc. All rights reserved.



Netflow v9 for IPv6

First Published: February 27, 2007

Last Updated: February 27, 2007

This module contains information about and instructions for configuring NetFlow and NetFlow Data Export (NDE) for capturing and exporting data from IP version 6 (IPv6) traffic flows using the NetFlow version 9 (v9) export format.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Netflow v9 for IPv6” section on page 9](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Prerequisites for Netflow v9 for IPv6, page 2](#)
- [Information About Netflow v9 for IPv6, page 2](#)
- [How to Configure Netflow v9 for IPv6, page 4](#)
- [Configuration Examples for Configuring Netflow v9 for IPv6, page 6](#)
- [Additional References, page 6](#)
- [Command Reference, page 8](#)
- [Feature Information for Netflow v9 for IPv6, page 9](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2007 Cisco Systems, Inc. All rights reserved.

Prerequisites for Netflow v9 for IPv6

Your router must be running Cisco IOS release 12.2(33)SRB or later to configure the Netflow v9 for IPv6 feature.

Information About Netflow v9 for IPv6

Before you configure the Netflow v9 for IPv6 feature, you should understand the following concepts:

- [NetFlow and NDE on the PFC, page 2](#)
- [NetFlow Export Format Version 9, page 2](#)

NetFlow and NDE on the PFC

The NetFlow cache on the PFC captures statistics for flows routed in hardware.

The PFC uses one of these flow masks to create NetFlow entries:

- **source-only**—The cache contains one entry for each source IP address. All flows from a given source IP address use this entry.
- **destination**—The cache contains one entry for each destination IP address. All flows to a given destination IP address use this entry.
- **destination-source**—The cache contains one entry for each source and destination IP address pair. All flows between the same source and destination IP addresses use this entry.
- **destination-source-interface**—Adds the source VLAN SNMP ifIndex to the information in the **destination-source** flow mask.
- **full**—A separate cache entry is created for each IP flow. A full entry includes the source IP address, destination IP address, protocol, and protocol interfaces.
- **full-interface**—Adds the source VLAN SNMP ifIndex to the information in the **full** flow mask.

See the “[Configuring NetFlow and NDE](#)” chapter of the *Cisco 7600 Series Cisco IOS Software Configuration Guide*, Release 12.2SR, for detailed information on NetFlow flow masks and flow records.

NetFlow Export Format Version 9

For all NetFlow export versions, the NetFlow export datagram consists of a header and a sequence of flow records. The header contains information such as sequence number, record count, and system uptime. The flow record contains flow information, such as IP addresses, ports, and routing information.

NetFlow version 9 export format is the newest NetFlow export format. The distinguishing feature of the NetFlow version 9 export format is that it is template based. Templates make the record format extensible. NetFlow version 9 export format allows future enhancements to NetFlow without requiring concurrent changes to the basic flow-record format.

The NetFlow version 9 export record format is different from the traditional NetFlow fixed format export record. In NetFlow version 9, a template describes the NetFlow data, and the flow set contains the actual data. This arrangement allows for flexible export.

The use of templates with the NetFlow version 9 export format provides several other key benefits:

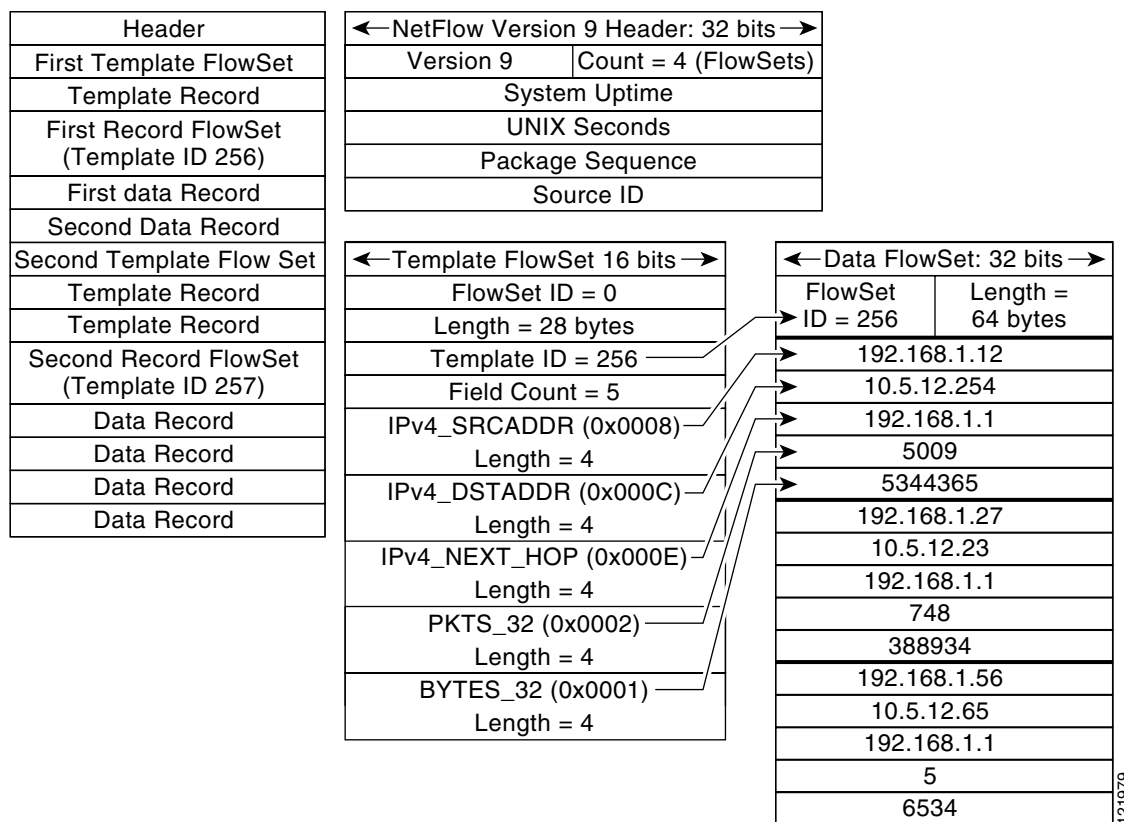
- You can export almost any information from a router or switch, including Layer 2 through 7 information, routing information, IP version 6 (IPv6), IP version 4 (IPv4), multicast, and Multiprotocol Label Switching (MPLS) information. This new information allows new applications for export data and new views of network behavior.
- Third-party business partners who produce applications that provide NetFlow collector or display services for NetFlow are not required to recompile their applications each time a new NetFlow export field is added. Instead, they can use an external data file that documents the known template formats.
- New features can be added to NetFlow more quickly, without breaking current implementations.
- NetFlow is “future-proofed” against new or developing protocols, because the version 9 export format can be adapted to provide support for them and for other non-NetFlow-based approaches to data collection.

The NetFlow version 9 export packet header format is shown in [Table 1](#).

Table 1 *NetFlow Version 9 Export Packet Header Field Names and Descriptions*

Bytes	Field Name	Description
0–1	Version	The version of NetFlow records exported in this packet; for version 9, this value is 0x0009.
2–3	Count	Number of FlowSet records (both template and data) contained within this packet.
4–7	System Uptime	Time in milliseconds since this device was first booted.
8–11	UNIX Seconds	Seconds since 0000 Coordinated Universal Time (UTC) 1970.
12–15	Sequence Number	Incremental sequence counter of all export packets sent by this export device; this value is cumulative, and it can be used to find out whether any export packets have been missed. This is a change from the NetFlow version 5 and version 8 headers, where this number represented “total flows.”
16–19	Source ID	The Source ID field is a 32-bit value that is used to guarantee uniqueness for each flow exported from a particular device. (The Source ID field is the equivalent of the engine type and engine ID fields found in the NetFlow version 5 and version 8 headers.) The format of this field is vendor specific. In Cisco’s implementation, the first two bytes are reserved for future expansion and are always zero. Byte 3 provides uniqueness with respect to the routing engine on the exporting device. Byte 4 provides uniqueness with respect to the particular line card or Versatile Interface Processor on the exporting device. Collector devices should use the combination of the source IP address and the Source ID field to associate an incoming NetFlow export packet with a unique instance of NetFlow on a particular device.

[Figure 1](#) shows a typical example of exporting data using the NetFlow version 9 export format.

Figure 1 NetFlow Version 9 Export Format Packet Example

Additional information about the NetFlow export format version 9 and the export format architecture is available in the [NetFlow version 9 Flow-Record Format](#) document.

How to Configure Netflow v9 for IPv6

Perform the steps in this required task to configure the Netflow v9 for IPv6 feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **mls flow {ip | ipv6} {destination | destination-source | full | interface-destination-source | interface-full | source}**
5. **mls nde sender**
6. **ip flow-export version 9**
7. **ip flow-export destination {ip-address | hostname} udp-port**
8. **interface type number**
9. **ipv6 address ip-address/mask**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	mls flow {ip ipv6} {destination destination-source full interface-destination-source interface-full source} Example: Router(config)# mls flow ipv6 interface-full	Specifies the NetFlow flow mask for IPv6 traffic.
Step 5	mls nde sender Example: Route(config)# mls nde sender	Enables NDE globally on the router. <p>Note NDE does not start exporting data until you specify a destination for the exported traffic. The destination for exported traffic is specified in Step 7.</p>
Step 6	ip flow-export version 9 Example: Router(config)# ip flow-export version 9	Configures NDE to use the NetFlow version 9 export format.
Step 7	ip flow-export destination {ip-address hostname} udp-port Example: Router(config)# ip flow-export destination 172.16.10.2 88	Specifies the IP address or the hostname of the NetFlow collector and the UDP port on which the NetFlow collector is listening.
Step 8	interface type number Example: Router(config)# interface fastethernet 1/1	Specifies the interface that you want to enable NetFlow on and enters interface configuration mode.
Step 9	ipv6 address ip-address/mask Example: Router(config-if)# ipv6 address 2001:0DB8:AB::2/64	Configure an IPv6 address on the interface.

Examples

The following output of the **show mls nde** command verifies that NDE is enabled on the router.

```
Router# show mls nde

NetFlow Data Export enabled
Exporting flows to 10.30.30.2 (12345) 172.16.10.2 (88)
Exporting flows from 10.4.9.149 (58970)
Version: 9
Layer2 flow creation is disabled
Layer2 flow export is disabled
Include Filter not configured
Exclude Filter not configured
Total NetFlow Data Export Packets are:
    0 packets, 0 no packets, 0 records
Total NetFlow Data Export Send Errors:
    IPWRITE_NO_FIB = 0
    IPWRITE_ADJ_FAILED = 0
    IPWRITE_PROCESS = 0
    IPWRITE_ENQUEUE_FAILED = 0
    IPWRITE_IPC_FAILED = 0
    IPWRITE_OUTPUT_FAILED = 0
    IPWRITE_MTU_FAILED = 0
    IPWRITE_ENCAPFIX_FAILED = 0
NetFlow Aggregation Disabled
```

Configuration Examples for Configuring Netflow v9 for IPv6

This section contains the following configuration example:

- [Configuring the NetFlow v9 for IPv6 Feature: Example, page 6](#)

Configuring the NetFlow v9 for IPv6 Feature: Example

The following example shows how to configure the router for NetFlow and NDE for IPv6 traffic using NetFlow export format version 9.

```
ipv6 unicast-routing
mls flow ipv6 interface-full
mls nde sender
ip flow-export version 9
ip flow-export destination 172.16.10.2 88
interface FastEthernet1/1
ipv6 address 2001:0DB8::1/64
```

Additional References

The following sections provide references related to the Netflow v9 for IPv6 feature.

Related Documents

Related Topic	Document Title
Platform-independent NetFlow commands, complete command syntax, command mode, defaults, command history, usage guidelines, and examples.	Cisco IOS NetFlow Command Reference , Release 12.2SR
Command reference for Cisco 7600 series routers	Cisco 7600 Series Cisco IOS Command Reference , Release 12.2SR

Standards

Standard	Title
There are no standards associated with this feature.	—

MIBs

MIB	MIBs Link
There are no MIBs associated with this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3954	Cisco Systems NetFlow Services Export Version 9

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

This feature uses no new or modified commands.

Feature Information for Netflow v9 for IPv6

Table 2 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Netflow v9 for IPv6

Feature Name	Releases	Feature Information
Netflow v9 for IPv6	12.2(33)SRB	The Netflow v9 for IPv6 feature enables the export of NetFlow flow information for IPv6 traffic. In 12.2(33)SRB, support for this feature was introduced on the Cisco 7600 series routers.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing OSPF for IPv6

First Published: March 17, 2003

Last Updated: August 18, 2008

The *Implementing OSPF for IPv6* module expands on Open Shortest Path First (OSPF) to provide support for IPv6 routing prefixes. This module describes the concepts and tasks you need to implement OSPF for IPv6 on your network.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing OSPF for IPv6” section on page 35](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Finding Feature Information, page 1](#)
- [Prerequisites for Implementing OSPF for IPv6, page 2](#)
- [Restrictions for Implementing OSPF for IPv6, page 2](#)
- [Information About Implementing OSPF for IPv6, page 2](#)
- [How to Implement OSPF for IPv6, page 11](#)
- [Configuration Examples for Implementing OSPF for IPv6, page 29](#)
- [Additional References, page 30](#)
- [Command Reference, page 32](#)
- [Feature Information for Implementing OSPF for IPv6, page 35](#)



Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2003–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing OSPF for IPv6

Before you enable OSPF for IPv6 on an interface, you must do the following:

- Complete the OSPF network strategy and planning for your IPv6 network. For example, you must decide whether multiple areas are required.
- Enable IPv6 unicast routing.
- Enable IPv6 on the interface.
- Configure the IP Security (IPsec) secure socket application program interface (API) on OSPF for IPv6 in order to enable authentication and encryption.

This document assumes that you are familiar with IPv4. Refer to the publications referenced in the [“Related Documents”](#) section for IPv4 configuration and command reference information.

Restrictions for Implementing OSPF for IPv6

- When running a dual-stack IP network with OSPF version 2 for IPv4 and OSPF for IPv6, be careful when changing the defaults for commands used to enable OSPF for IPv6. Changing these defaults may affect your OSPF for IPv6 network, possibly adversely.
- Authentication is supported as of Cisco IOS Release 12.3(4)T.
- ESP authentication and encryption are supported as of Cisco IOS Release 12.4(9)T.
- A packet will be rejected on a router if the packet is coming from an IPv6 address that is found on any interface on the same router.

Information About Implementing OSPF for IPv6

To implement OSPF for IPv6, you need to understand the following concepts:

- [How OSPF for IPv6 Works, page 3](#)
- [Comparison of OSPF for IPv6 and OSPF Version 2, page 3](#)
- [LSA Types for IPv6, page 3](#)
- [Force SPF in OSPF for IPv6, page 5](#)
- [Fast Convergence—LSA and SPF Throttling, page 5](#)
- [Load Balancing in OSPF for IPv6, page 5](#)
- [Importing Addresses into OSPF for IPv6, page 6](#)
- [OSPF for IPv6 Customization, page 6](#)
- [OSPF for IPv6 Authentication Support with IPsec, page 6](#)
- [Link Quality Metrics Reporting for OSPFv3 with VMI Interfaces, page 7](#)
- [OSPFv3 Graceful Restart, page 10](#)

How OSPF for IPv6 Works

OSPF is a routing protocol for IP. It is a link-state protocol, as opposed to a distance-vector protocol. Think of a link as being an interface on a networking device. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines. The state of a link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network it is connected to, the routers connected to that network, and so on. This information is propagated in various type of link-state advertisements (LSAs).

A router's collection of LSA data is stored in a link-state database. The contents of the database, when subjected to the Dijkstra algorithm, result in the creation of the OSPF routing table. The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations via specific router interface ports.

OSPF version 3, which is described in RFC 2740, supports IPv6.

Comparison of OSPF for IPv6 and OSPF Version 2

Much of the OSPF for IPv6 feature is the same as in OSPF version 2. OSPF version 3 for IPv6, which is described in RFC 2740, expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.

In OSPF for IPv6, a routing process does not need to be explicitly created. Enabling OSPF for IPv6 on an interface will cause a routing process, and its associated configuration, to be created.

In OSPF for IPv6, each interface must be enabled using commands in interface configuration mode. This feature is different from OSPF version 2, in which interfaces are indirectly enabled using the router configuration mode.

When using a nonbroadcast multiaccess (NBMA) interface in OSPF for IPv6, users must manually configure the router with the list of neighbors. Neighboring routers are identified by their router ID.

In IPv6, users can configure many address prefixes on an interface. In OSPF for IPv6, all address prefixes on an interface are included by default. Users cannot select some address prefixes to be imported into OSPF for IPv6; either all address prefixes on an interface are imported, or no address prefixes on an interface are imported.

Unlike OSPF version 2, multiple instances of OSPF for IPv6 can be run on a link.

In OSPF for IPv6, it is possible that no IPv4 addresses will be configured on any interface. In this case, the user must use the **router-id** command to configure a router ID before the OSPF process will be started. A router ID is a 32-bit opaque number. OSPF version 2 takes advantage of the 32-bit IPv4 address to pick an IPv4 address as the router ID. If an IPv4 address does exist when OSPF for IPv6 is enabled on an interface, then that IPv4 address is used for the router ID. If more than one IPv4 address is available, a router ID is chosen using the same rules as for OSPF version 2.

OSPF automatically prefers a loopback interface over any other kind, and it chooses the highest IP address among all loopback interfaces. If no loopback interfaces are present, the highest IP address in the router is chosen. You cannot tell OSPF to use any particular interface.

LSA Types for IPv6

The following list describes LSA types, each of which has a different purpose:

- Router LSAs (Type 1)—Describes the link state and costs of a router's links to the area. These LSAs are flooded within an area only. The LSA indicates if the router is an Area Border Router (ABR) or Autonomous System Boundary Router (ASBR), and if it is one end of a virtual link. Type 1 LSAs are also used to advertise stub networks. In OSPF for IPv6, these LSAs have no address information and are network-protocol-independent. In OSPF for IPv6, router interface information may be spread across multiple router LSAs. Receivers must concatenate all router LSAs originated by a given router when running the SPF calculation.
- Network LSAs (Type 2)—Describes the link-state and cost information for all routers attached to the network. This LSA is an aggregation of all the link-state and cost information in the network. Only a designated router tracks this information and can generate a network LSA. In OSPF for IPv6, network LSAs have no address information and are network-protocol-independent.
- Interarea-prefix LSAs for ABRs (Type 3)—Advertises internal networks to routers in other areas (interarea routes). Type 3 LSAs may represent a single network or a set of networks summarized into one advertisement. Only ABRs generate summary LSAs. In OSPF for IPv6, addresses for these LSAs are expressed as *prefix, prefix length* instead of *address, mask*. The default route is expressed as a prefix with length 0.
- Interarea-router LSAs for ASBRs (Type 4)—Advertise the location of an ASBR. Routers that are trying to reach an external network use these advertisements to determine the best path to the next hop. ASBRs generate Type 4 LSAs.
- Autonomous system external LSAs (Type 5)—Redistributes routes from another AS, usually from a different routing protocol into OSPF. In OSPF for IPv6, addresses for these LSAs are expressed as *prefix, prefix length* instead of *address, mask*. The default route is expressed as a prefix with length 0.
- Link LSAs (Type 8)—Have local-link flooding scope and are never flooded beyond the link with which they are associated. Link LSAs provide the link-local address of the router to all other routers attached to the link, inform other routers attached to the link of a list of IPv6 prefixes to associate with the link, and allow the router to assert a collection of Options bits to associate with the network LSA that will be originated for the link.
- Intra-Area-Prefix LSAs (Type 9)—A router can originate multiple intra-area-prefix LSAs for each router or transit network, each with a unique link-state ID. The link-state ID for each intra-area-prefix LSA describes its association to either the router LSA or the network LSA and contains prefixes for stub and transit networks.

An address prefix occurs in almost all newly defined LSAs. The prefix is represented by three fields: PrefixLength, PrefixOptions, and Address Prefix. In OSPF for IPv6, addresses for these LSAs are expressed as *prefix, prefix length* instead of *address, mask*. The default route is expressed as a prefix with length 0. Type 3 and Type 9 LSAs carry all IPv6 prefix information that, in IPv4, is included in router LSAs and network LSAs. The Options field in certain LSAs (router LSAs, network LSAs, interarea-router LSAs, and link LSAs) has been expanded to 24 bits to provide support for OSPF in IPv6.

In OSPF for IPv6, the sole function of link-state ID in interarea-prefix LSAs, interarea-router LSAs, and autonomous-system external LSAs is to identify individual pieces of the link-state database. All addresses or router IDs that are expressed by the link-state ID in OSPF version 2 are carried in the body of the LSA in OSPF for IPv6.

The link-state ID in network LSAs and link LSAs is always the interface ID of the originating router on the link being described. For this reason, network LSAs and link LSAs are now the only LSAs whose size cannot be limited. A network LSA must list all routers connected to the link, and a link LSA must list all of the address prefixes of a router on the link.

NBMA in OSPF for IPv6

On NBMA networks, the designated router (DR) or backup DR (BDR) performs the LSA flooding. On point-to-point networks, flooding simply goes out an interface directly to a neighbor.

Routers that share a common segment (Layer 2 link between two interfaces) become neighbors on that segment. OSPF uses the Hello protocol, periodically sending hello packets out each interface. Routers become neighbors when they see themselves listed in the neighbor's hello packet. After two routers become neighbors, they may proceed to exchange and synchronize their databases, which creates an adjacency. Not all neighboring routers have an adjacency.

On point-to-point and point-to-multipoint networks, the software floods routing updates to immediate neighbors. There is no DR or BDR; all routing information is flooded to each networking device.

On broadcast or NBMA segments only, OSPF minimizes the amount of information being exchanged on a segment by choosing one router to be a DR and one router to be a BDR. Thus, the routers on the segment have a central point of contact for information exchange. Instead of each router exchanging routing updates with every other router on the segment, each router exchanges information with the DR and BDR. The DR and BDR relay the information to the other routers.

The software looks at the priority of the routers on the segment to determine which routers will be the DR and BDR. The router with the highest priority is elected the DR. If there is a tie, then the router with the higher router ID takes precedence. After the DR is elected, the BDR is elected the same way. A router with a router priority set to zero is ineligible to become the DR or BDR.

When using NBMA in OSPF for IPv6, you cannot automatically detect neighbors. On an NBMA interface, you must configure your neighbors manually using interface configuration mode.

Force SPF in OSPF for IPv6

When the **process** keyword is used with the **clear ipv6 ospf** command, the OSPF database is cleared and repopulated, and then the SPF algorithm is performed. When the **force-spf** keyword is used with the **clear ipv6 ospf** command, the OSPF database is not cleared before the SPF algorithm is performed.

Fast Convergence—LSA and SPF Throttling

The OSPF for IPv6 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPF during times of network instability. It also allows faster OSPF convergence by providing LSA rate limiting in milliseconds.

Previously, OSPF for IPv6 used static timers for rate-limiting SPF calculation and LSA generation. Although these timers are configurable, the values used are specified in seconds, which poses a limitation on OSPF for IPv6 convergence. LSA and SPF throttling achieves subsecond convergence by providing a more sophisticated SPF and LSA rate-limiting mechanism that is able to react quickly to changes and also provide stability and protection during prolonged periods of instability.

Load Balancing in OSPF for IPv6

When a router learns multiple routes to a specific network via multiple routing processes (or routing protocols), it installs the route with the lowest administrative distance in the routing table. Sometimes the router must select a route from among many learned via the same routing process with the same

administrative distance. In this case, the router chooses the path with the lowest cost (or metric) to the destination. Each routing process calculates its cost differently and the costs may need to be manipulated in order to achieve load balancing.

OSPF performs load balancing automatically in the following way. If OSPF finds that it can reach a destination through more than one interface and each path has the same cost, it installs each path in the routing table. The only restriction on the number of paths to the same destination is controlled by the **maximum-paths** command. The default maximum paths is 16, and the range is from 1 to 64.

Importing Addresses into OSPF for IPv6

When importing the set of addresses specified on an interface on which OSPF for IPv6 is running into OSPF for IPv6, users cannot select specific addresses to be imported. Either all addresses are imported, or no addresses are imported.

OSPF for IPv6 Customization

You can customize OSPF for IPv6 for your network, but you likely will not need to do so. The defaults for OSPF in IPv6 are set to meet the requirements of most customers and features. If you must change the defaults, refer to the IPv6 command reference to find the appropriate syntax.



Caution

Be careful when changing the defaults. Changing defaults will affect your OSPF for IPv6 network, possibly adversely.

OSPF for IPv6 Authentication Support with IPsec

In order to ensure that OSPF for IPv6 packets are not altered and re-sent to the router, causing the router to behave in a way not desired by its managers, OSPF for IPv6 packets must be authenticated. OSPF for IPv6 uses the IP Security (IPsec) secure socket application program interface (API) to add authentication to OSPF for IPv6 packets. This API has been extended to provide support for IPv6.

OSPF for IPv6 requires the use of IPsec to enable authentication. Crypto images are required to use authentication, because only crypto images include the IPsec API needed for use with OSPF for IPv6.

In OSPF for IPv6, authentication fields have been removed from OSPF headers. When OSPF runs on IPv6, OSPF requires the IPv6 authentication header (AH) or IPv6 ESP header to ensure integrity, authentication, and confidentiality of routing exchanges. IPv6 AH and ESP extension headers can be used to provide authentication and confidentiality to OSPF for IPv6.

To use the IPsec AH, you must enable the **ipv6 ospf authentication** command. To use the IPsec ESP, you must enable the **ipv6 ospf encryption** command. The ESP header may be applied alone or in combination with the AH, and when ESP is used, both encryption and authentication are provided. Security services can be provided between a pair of communicating hosts, between a pair of communicating security gateways, or between a security gateway and a host.

To configure IPsec, users configure a security policy, which is a combination of the security policy index (SPI) and the key (the key is used to create and validate the hash value). IPsec for OSPF for IPv6 can be configured on an interface or on an OSPF area. For higher security, users should configure a different policy on each interface configured with IPsec. If a user configures IPsec for an OSPF area, the policy is applied to all of the interfaces in that area, except for the interfaces that have IPsec configured directly. Once IPsec is configured for OSPF for IPv6, IPsec is invisible to the user.

The secure socket API is used by applications to secure traffic. The API needs to allow the application to open, listen, and close secure sockets. The binding between the application and the secure socket layer also allows the secure socket layer to inform the application of changes to the socket, such as connection open and close events. The secure socket API is able to identify the socket; that is, it can identify the local and remote addresses, masks, ports, and protocol that carry the traffic requiring security.

Each interface has a secure socket state, which can be one of the following:

- **NULL:** Do not create a secure socket for the interface if authentication is configured for the area.
- **DOWN:** IPsec has been configured for the interface (or the area that contains the interface), but OSPF for IPv6 either has not requested IPsec to create a secure socket for this interface, or there is an error condition.
- **GOING UP:** OSPF for IPv6 has requested a secure socket from IPsec and is waiting for a CRYPTO_SS_SOCKET_UP message from IPsec.
- **UP:** OSPF has received a CRYPTO_SS_SOCKET_UP message from IPsec.
- **CLOSING:** The secure socket for the interface has been closed. A new socket may be opened for the interface, in which case the current secure socket makes the transition to the DOWN state. Otherwise, the interface will become UNCONFIGURED.
- **UNCONFIGURED:** Authentication is not configured on the interface.

OSPF will not send or accept packets while in the DOWN state.

For further information on IPsec, refer to the [Implementing IPsec in IPv6 Security](#) document.

OSPF for IPv6 Virtual Links

For each virtual link, a master security information datablock is created for the virtual link. Because a secure socket must be opened on each interface, there will be a corresponding security information datablock for each interface in the transit area. The secure socket state is kept in the interface's security information datablock. The state field in the master security information datablock reflects the status of all of the secure sockets opened for the virtual link. If all of the secure sockets are UP, then the security state for the virtual link will be set to UP.

Packets sent on a virtual link with IPsec must use predetermined source and destination addresses. The first local area address found in the router's intra-area-prefix LSA for the area is used as the source address. This source address is saved in the area data structure and used when secure sockets are opened and packets sent over the virtual link. The virtual link will not transition to the point-to-point state until a source address is selected. Also, when the source or destination address changes, the previous secure sockets must be closed and new secure sockets opened.

Link Quality Metrics Reporting for OSPFv3 with VMI Interfaces

OSPFv3 is one of the routing protocols that can be used with Virtual Multipoint Interfaces (VMIs) in router-to-radio networks. The quality of a radio link has a direct impact on the throughput that can be achieved by router-router traffic. The PPPoE protocol has been extended to provide a process by which a router can request, or a radio can report, link quality metric information. Cisco's OSFPv3 implementation has been enhanced so that the route cost to a neighbor is dynamically updated based on metrics reported by the radio, thus allowing the best route to be chosen within a given set of radio links.

The routing protocols receive raw radio link data, and compute a composite quality metric for each link. In computing these metrics, the following factors may be considered:

- **Maximum Data Rate**—the theoretical maximum data rate of the radio link, in bytes per second

- Current Data Rate—the current data rate achieved on the link, in bytes per second
- Latency—the transmission delay packets encounter, in milliseconds
- Resources—a percentage (0 to 100) that can represent the remaining amount of a resource (such as battery power)
- Relative Link Quality—a numeric value (0-100) representing relative quality, with 100 being the highest quality

Metrics can be weighted during the configuration process to emphasize or de-emphasize particular characteristics. For example, if throughput is a particular concern, the current data rate metric could be weighted so that it is factored more heavily into the composite metric. Similarly, a metric that is of no concern can be omitted from the composite calculation.

Link metrics can change rapidly, often by very small degrees, which could result in a flood of meaningless routing updates. In a worst case scenario, the network would be churning almost continuously as it struggled to react to minor variations in link quality. To alleviate this concern, Cisco provides a tunable dampening mechanism that allows the user to configure threshold values. Any metric change that falls below the threshold is ignored. The quality of a connection to a neighbor varies, based on various characteristics of the interface when OSPF is used as the routing protocol. The routing protocol receives dynamic raw radio link characteristics and computes a composite metric that is used to reduce the effect of frequent routing changes.

A tunable hysteresis mechanism allows users to adjust the threshold to the routing changes that occur when the router receives a signal that a new peer has been discovered, or that an existing peer is unreachable. The tunable metric is weighted and is adjusted dynamically to account for the following characteristics:

- Current and maximum bandwidth
- Latency
- Resources
- L2 factor

Individual weights can be deconfigured and all weights can be cleared so that the cost is set back to the default value for the interface type. Based on the routing changes that occur, cost can be determined by the application of these metrics.

OSPF Cost Calculation

Because cost components can change rapidly, it might be necessary to dampen the volume of changes to reduce network-wide churn. The recommended values for S2, S3, and S4 are based on network simulations that may reduce the rate of network changes. The recommended value for S1 is zero to eliminate this variable from the route cost calculation.

The overall link cost is computed using the following formula shown in [Figure 1](#).

Figure 1 Overall Link Cost Formula

$$\text{LinkCost} = \text{OC} + \text{BW} \left(\frac{\text{Throughput_weight}}{100} \right) + \text{Resources} \left(\frac{\text{Resources_weight}}{100} \right) + \text{Latency} \left(\frac{\text{Latency_weight}}{100} \right) + \text{L2_factor} \left(\frac{\text{L2_weight}}{100} \right)$$

$$\text{OC} = \left[\frac{(\text{ospf_reference_bw})(1 \times 10^8)}{(\text{MDR})(1024)} \right]$$

Note: The default ospf reference bw is 100

$$\text{BW} = \frac{(65535 + 1) \left(100 - \frac{\text{CDR}}{\text{MDR}} (100) \right)}{100}$$

$$\text{Resources} = \frac{(100 - \text{resources})^3 (65536)}{100}$$

$$\text{Latency} = \text{latency}$$

$$\text{L2_factor} = \frac{(100 - \text{RLF})(65535 + 1)}{100}$$

2310-48

Table 1 defines the symbols used in the OSPF cost calculation.

Table 1 OSPF Cost Calculation Definitions

Cost Component	Component Definition
OC	The “default OSPF cost.” Calculated from reference bandwidth using reference_bw / (MDR*1000), where reference_bw=10^8
A through D	Various radio-specific data based formulas that produce results in the 0 through 64k range.
A	CDR- and MDR-related formula: $(2^{16} * (100 - (\text{CDR} * 100 / \text{MDR}))) / 100$
B	Resources related formula: $((100 - \text{RESOURCES})^3 * 2^{16} / 10^6)$
C	Latency as reported by the radio, already in the 0 through 64K range when reported (LATENCY).
D	RLF-related formula: $((100 - \text{RLF}) * 2^{16}) / 100$
S1 through S4	Scalar weighting factors input from CLI. These scalars scale down the values as computed by A through D. The value of 0 disables and value of 100 enables full 0 through 64k range for one component.

While each network might have unique characteristics that require different settings to optimize actual network performance, these are recommended values intended as a starting point for optimizing a OSPFv3 network. Table 2 lists the recommended value settings for OSPF cost metrics.

Table 2 **Recommended Value Settings for OSPF Cost Metrics**

Setting	Metric Description	Default Value	Recommended Value
S1	ipv6 ospf dynamic weight throughout	100	0
S2	ipv6 ospf dynamic weight resources	100	29
S3	ipv6 ospf dynamic weight latency	100	29
S4	ipv6 ospf dynamic weight L2 factor	100	29

Using this formula, the default path costs were calculated as noted in the following list. If these values do not suit your network, you can use your own method of calculating path costs.

- 56-kbps serial link—Default cost is 1785.
- 64-kbps serial link—Default cost is 1562.
- T1 (1.544-Mbps serial link)—Default cost is 64.
- E1 (2.048-Mbps serial link)—Default cost is 48.
- 4-Mbps Token Ring—Default cost is 25.
- Ethernet—Default cost is 10.
- 16-Mbps Token Ring—Default cost is 6.
- FDDI—Default cost is 1.
- X25—Default cost is 5208.
- Asynchronous—Default cost is 10,000.
- ATM—Default cost is 1.

To illustrate these settings, the following example shows how OSPF cost metrics might be defined for a VMI interface:

```
interface vm1
  ipv6 ospf cost dynamic weight throughout 0
  ipv6 ospf cost dynamic weight resources 29
  ipv6 ospf cost dynamic weight latency 29
  ipv6 ospf cost dynamic weight L2-factor 29
```

OSPFv3 Graceful Restart

The graceful restart feature in OSPFv3 allows nonstop data forwarding along routes that are already known while the OSPFv3 routing protocol information is being restored. A router can participate in graceful restart either in restart mode (such as in a graceful-restart-capable router) or in helper mode (such as in a graceful-restart-aware router).

To perform the graceful restart function, a router must be in high availability (HA) stateful switchover (SSO) mode (that is, dual RP). A router capable of graceful restart will perform the graceful restart function when the following failures occur:

- A Route Processor (RP) failure that results in switchover to standby RP
- A planned RP switchover to standby RP

The graceful restart feature requires that neighboring routers be graceful-restart aware.

For further information about SSO and nonstop forwarding (NSF), see the [Stateful Switchover](#) and [Cisco Nonstop Forwarding](#) documents.

How to Implement OSPF for IPv6

This section contains the following procedures:

- [Enabling OSPF for IPv6 on an Interface, page 11](#) (required)
- [Defining an OSPF for IPv6 Area Range, page 12](#) (optional)
- [Configuring IPsec on OSPF for IPv6, page 13](#) (optional)
- [Configuring NBMA Interfaces, page 18](#) (optional)
- [Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence, page 19](#) (optional)
- [Enabling OSPFv3 Graceful Restart, page 22](#) (optional)
- [Forcing an SPF Calculation, page 24](#) (optional)
- [Verifying OSPF for IPv6 Configuration and Operation, page 24](#) (optional)

Enabling OSPF for IPv6 on an Interface

This task explains how to enable OSPF for IPv6 routing and configure OSPF for IPv6 on each interface. By default, OSPF for IPv6 routing is disabled and OSPF for IPv6 is not configured on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 ospf process-id area area-id [instance instance-id] Example: Router(config-if)# ipv6 ospf 1 area 0	Enables OSPF for IPv6 on an interface.

Defining an OSPF for IPv6 Area Range

The cost of the summarized routes will be the highest cost of the routes being summarized. For example, if the following routes are summarized:

```
OI 2001:0DB8:0:0:7::/64 [110/20]
    via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2001:0DB8:0:0:8::/64 [110/100]
    via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2001:0DB8:0:0:9::/64 [110/20]
    via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
```

They become one summarized route, as follows:

```
OI 2001:0DB8::/48 [110/100]
    via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
```

This task explains how to consolidate or summarize routes for an OSPF area.

Prerequisites

OSPF for IPv6 routing must be enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf process-id**
4. **area area-id range ipv6-prefix/prefix-length [advertise | not-advertise] [cost cost]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router ospf process-id Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	area area-id range ipv6-prefix/prefix-length [advertise not-advertise] [cost cost] Example: Router(config-rtr)# area 1 range 2001:0DB8::/48	Consolidates and summarizes routes at an area boundary.

Configuring IPsec on OSPF for IPv6

Once you have configured OSPF for IPv6 and decided on your authentication, you must define the security policy on each of the routers within the group. The security policy consists of the combination of the key and the SPI. To define a security policy, you must define an SPI and a key.

You can configure an authentication or encryption policy either on an interface or for an OSPF area. When you configure for an area, the security policy is applied to all of the interfaces in the area. For higher security, use a different policy on each interface.

You can configure authentication and encryption on virtual links.

The following tasks explain how to configure authentication and encryption on an interface or in an OSPF area, and on virtual links.

- [Defining Authentication on an Interface, page 13](#)
- [Defining Encryption on an Interface, page 14](#)
- [Defining Authentication in an OSPF Area, page 15](#)
- [Defining Encryption in an OSPF Area, page 16](#)
- [Defining Authentication and Encryption for a Virtual Link in an OSPF Area, page 17](#)

Defining Authentication on an Interface

This task explains how to define authentication on an interface.

Prerequisites

Before you configure IPsec on an interface, you must configure OSPF for IPv6 on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 ospf authentication ipsec spi** *spi* **md5** [*key-encryption-type* {*key* | **null**}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 ospf authentication ipsec spi <i>spi</i> md5 [<i>key-encryption-type</i> { <i>key</i> null }] Example: Router(config-if)# ipv6 ospf authentication ipsec spi 500 md5 1234567890abcdef1234567890abcdef	Specifies the authentication type for an interface.

Defining Encryption on an Interface

This task describes how to define encryption on an interface.

Prerequisites

Before you configure IPsec on an interface, you must configure OSPF for IPv6 on that interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **ipv6 ospf encryption** {**ipsec spi spi esp encryption-algorithm** [[*key-encryption-type*] *key*] *authentication-algorithm* [*key-encryption-type*] *key* | **null**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	ipv6 ospf encryption { ipsec spi spi esp encryption-algorithm [[<i>key-encryption-type</i>] <i>key</i>] <i>authentication-algorithm</i> [<i>key-encryption-type</i>] <i>key</i> null }	Specifies the encryption type for an interface.
	Example: Router(config-if) ipv6 ospf encryption ipsec spi 1001 esp null sha1 123456789A123456789B123456789C123456789D	

Defining Authentication in an OSPF Area

This task explains how to define authentication in an OSPF area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **area** *area-id* **authentication ipsec spi spi md5** [*key-encryption-type*] *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 router ospf process-id	Enables OSPF router configuration mode.
	Example: Router(config)# ipv6 router ospf 1	
Step 4	area area-id authentication ipsec spi spi md5 [key-encryption-type] key	Enables authentication in an OSPF area.
	Example: Router(config-rtr)# area 1 authentication ipsec spi 678 md5 1234567890ABCDEF1234567890ABCDEF	

Defining Encryption in an OSPF Area

This task describes how to define encryption in an OSPF area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf process-id**
4. **area area-id encryption ipsec spi spi esp encryption-algorithm** [[key-encryption-type] key]
authentication-algorithm [key-encryption-type] key

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	ipv6 router ospf <i>process-id</i> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	area <i>area-id</i> encryption ipsec spi spi esp <i>encryption-algorithm</i> <i>[[key-encryption-type]</i> <i>key]</i> <i>authentication-algorithm</i> <i>[[key-encryption-type]</i> <i>key</i> Example: Router(config-rtr)# area 1 encryption ipsec spi 500 esp null md5 1aaa2bbb3ccc4ddd5eee6fff7aaa8bbb	Enables encryption in an OSPF area.

Defining Authentication and Encryption for a Virtual Link in an OSPF Area

The following task describes how to define authentication and encryption for virtual links in an OSPF area.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **area** *area-id* **virtual-link** *router-id* **authentication ipsec spi spi authentication-algorithm** *[[key-encryption-type]* *key*
5. **area** *area-id* **virtual-link** *router-id* **encryption ipsec spi spi esp encryption-algorithm** *[[key-encryption-type]* *key]* *authentication-algorithm* *[[key-encryption-type]* *key*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router ospf <i>process-id</i> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.

	Command or Action	Purpose
Step 4	<pre>area area-id virtual-link router-id authentication ipsec spi spi authentication-algorithm [key-encryption-type] key</pre> <p>Example:</p> <pre>Router(config-rtr)# area 1 virtual-link 10.0.0.1 authentication ipsec spi 940 md5 1234567890ABCDEF1234567890ABCDEF</pre>	Enables authentication for virtual links in an OSPF area.
Step 5	<pre>area area-id virtual-link router-id encryption ipsec spi spi esp encryption-algorithm [[key-encryption-type] key] authentication-algorithm [key-encryption-type] key</pre> <p>Example:</p> <pre>Router(config-rtr)# area 1 virtual-link 10.1.0.1 hello-interval 2 dead-interval 10 encryption ipsec spi 3944 esp null sha1 123456789A123456789B123456789C123456789D</pre>	Enables encryption for virtual links in an OSPF area.

Configuring NBMA Interfaces

You can customize OSPF for IPv6 in your network to use NBMA interfaces. OSPF for IPv6 cannot automatically detect neighbors over NBMA interfaces. On an NBMA interface, you must configure your neighbors manually using interface configuration mode. This task explains how to configure NBMA interfaces.

Prerequisites

Before you configure NBMA interfaces, you must perform the following tasks:

- Configure your network to be an NBMA network
- Identify each neighbor

Restrictions

- You cannot automatically detect neighbors when using NBMA interfaces. You must manually configure your router to detect neighbors when using an NBMA interface.
- When configuring the **ipv6 ospf neighbor** command, the IPv6 address used must be the link-local address of the neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. **frame-relay map ipv6** *ipv6-address dlci* [**broadcast**] [**cisco**] [**ietf**] [**payload-compression** {**packet-by-packet** | **frf9 stac** [*hardware-options*] | **data-stream stac** [*hardware-options*]}]
5. **ipv6 ospf neighbor** *ipv6-address* [**priority number**] [**poll-interval seconds**] [**cost number**] [**database-filter all out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface serial 0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	frame-relay map ipv6 <i>ipv6-address dlci</i> [broadcast] [cisco] [ietf] [payload-compression { packet-by-packet frf9 stac [<i>hardware-options</i>] data-stream stac [<i>hardware-options</i>]}]	Defines the mapping between a destination IPv6 address and the data-link connection identifier (DLCI) used to connect to the destination address. <ul style="list-style-type: none"> In this example, the NBMA link is frame relay. For other kinds of NBMA links, different mapping commands are used.
Step 5	ipv6 ospf neighbor <i>ipv6-address</i> [priority number] [poll-interval seconds] [cost number] [database-filter all out]	Configures an OSPF for IPv6 neighboring router.
	Example: Router(config-if) ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C01	

Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence

This task explains how to configure LSA and SPF throttling for the OSPF for IPv6 Fast Convergence feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*

4. **timers throttle spf** *spf-start spf-hold spf-max-wait*
5. **timers throttle lsa** *start-interval hold-interval max-interval*
6. **timers lsa arrival** *milliseconds*
7. **timers pacing flood** *milliseconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router ospf <i>process-id</i> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	timers throttle spf <i>spf-start spf-hold spf-max-wait</i> Example: Router(config-rtr)# timers throttle spf 200 200 200	Turns on SPF throttling.
Step 5	timers throttle lsa <i>start-interval hold-interval max-interval</i> Example: Router(config-rtr)# timers throttle lsa 300 300 300	Sets rate-limiting values for OSPF for IPv6 LSA generation.
Step 6	timers lsa arrival <i>milliseconds</i> Example: Router(config-rtr)# timers lsa arrival 300	Sets the minimum interval at which the software accepts the same LSA from OSPF neighbors.
Step 7	timers pacing flood <i>milliseconds</i> Example: Router(config-rtr)# timers pacing flood 30	Configures LSA flood packet pacing.

Enabling Event Logging for LSA and SPF Rate Limiting

An OSPF for IPv6 event log is kept for each OSPF for IPv6 instance. This task explains how to enable event logging for the LSA and SPF rate-limiting function.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf *process-id***
4. **event-log [size *[number of events]*] [one-shot] [pause]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router ospf <i>process-id</i> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	event-log [size <i>[number of events]</i>] [one-shot] [pause] Example: Router(config-rtr)# event-log size 10000 one-shot	Enables event logging.

Clearing the Content of an Event Log

This task explains how to clear an event log.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 ospf [*process-id*] events**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. Enter your password if prompted.
	Example: Router> enable	
Step 2	clear ipv6 ospf [<i>process-id</i>] events	Clears the OSPF for IPv6 event log content based on the OSPF routing process ID.
	Example: Router# clear ipv6 ospf 1 events	

Enabling OSPFv3 Graceful Restart

The graceful restart feature may be enabled on graceful-restart-capable routers and on graceful-restart-aware routers. The following sections describe how to enable OSPFv3 graceful restart:

- [Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Capable Router, page 22](#)
- [Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Aware Router, page 23](#)

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Capable Router

This task describes how to enable OSPFv3 graceful restart on a graceful-restart-capable router.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router ospf** *process-id*
4. **graceful-restart** [*restart-interval interval*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	<code>ipv6 router ospf process-id</code> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	<code>graceful-restart [restart-interval interval]</code> Example: Router(config-rtr)# graceful-restart	Enables the OSPFv3 graceful restart feature on a graceful-restart-capable router.

Enabling OSPFv3 Graceful Restart on a Graceful-Restart-Aware Router

This task describes how to enable OSPFv3 graceful restart on a graceful-restart-aware router.

SUMMARY STEPS

1. `enable`
2. `configure terminal`
3. `ipv6 router ospf process-id`
4. `graceful-restart helper {disable | strict-lsa-checking}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.
Step 3	<code>ipv6 router ospf process-id</code> Example: Router(config)# ipv6 router ospf 1	Enables OSPF router configuration mode.
Step 4	<code>graceful-restart helper {disable strict-lsa-checking}</code> Example: Router(config-rtr)# graceful-restart helper strict-lsa-checking	Enables the OSPFv3 graceful restart feature on a graceful-restart-aware router.

Forcing an SPF Calculation

This task explains how to start the SPF algorithm without first clearing the OSPF database.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 ospf** [*process-id*] {**process** | **force-spf** | **redistribution**}

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	clear ipv6 ospf [<i>process-id</i>] { process force-spf redistribution } Example: Router# clear ipv6 ospf force-spf	Clears the OSPF state based on the OSPF routing process ID, and forces the start of the SPF algorithm.

Verifying OSPF for IPv6 Configuration and Operation

This task explains how to display information to verify the configuration and operation of OSPF for IPv6.

SUMMARY STEPS

1. **enable**
2. **show ipv6 ospf** [*process-id*] [*area-id*] **interface** [*interface-type interface-number*]
3. **show ipv6 ospf** [*process-id*] [*area-id*]
4. **show crypto ipsec policy** [*name policy-name*]
5. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **interface** *type number* | **peer** [*vrf fvr-f-name*] **address** | **vrf** *ivrf-name* | **ipv6** [*interface-type interface-number*]] [**detail**]
6. **show ipv6 ospf** [*process-ID*] **event** [**generic** | **interface** | **lsa** | **neighbor** | **reverse** | **rib** | **spf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] interface [<i>interface-type interface-number</i>] Example: Router# show ipv6 ospf interface	Displays OSPF-related interface information.
Step 3	show ipv6 ospf [<i>process-id</i>] [<i>area-id</i>] Example: Router# show ipv6 ospf	Displays general information about OSPF routing processes.
Step 4	show crypto ipsec policy [<i>name policy-name</i>] Example: Router# show crypto ipsec policy	Displays the parameters for each IPsec parameter.
Step 5	show crypto ipsec sa [<i>map map-name</i> address identity interface type number peer vrf fvrf-name] address vrf ivrf-name ipv6 [<i>interface-type interface-number</i>] [detail] Example: Router# show crypto ipsec sa ipv6	Displays the settings used by current security associations (SAs).
Step 6	show ipv6 ospf [<i>process-ID</i>] event [generic interface lsa neighbor reverse rib spf] Example: Router# show ipv6 ospf event spf	Displays detailed information about OSPF for IPv6 events.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 ospf interface Command, page 25](#)
- [Sample Output from the show ipv6 ospf Command, page 27](#)
- [Sample Output from the show crypto ipsec policy Command, page 27](#)
- [Sample Output from the show crypto ipsec sa ipv6 Command, page 28](#)
- [Sample Output from the show ipv6 ospf graceful-restart Command, page 28](#)

Sample Output from the show ipv6 ospf interface Command

The following is sample output from the **show ipv6 ospf interface** command with regular interfaces and a virtual link that are protected by encryption and authentication:

Router# **show ipv6 ospf interface**

```

OSPFv3_VL1 is up, line protocol is up
  Interface ID 69
  Area 0, Process ID 1, Instance ID 0, Router ID 10.0.0.1
  Network Type VIRTUAL_LINK, Cost: 64
  Configured as demand circuit.
  Run as demand circuit.
  DoNotAge LSA allowed.
  NULL encryption SHA-1 auth SPI 3944, secure socket UP (errors: 0)
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 2, Dead 10, Wait 40, Retransmit 5
    Hello due in 00:00:00
  Index 1/3/5, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.0.1 (Hello suppressed)
  Suppress hello for 1 neighbor(s)
OSPFv3_VL0 is up, line protocol is up
  Interface ID 67
  Area 0, Process ID 1, Instance ID 0, Router ID 10.0.0.1
  Network Type VIRTUAL_LINK, Cost: 128
  Configured as demand circuit.
  Run as demand circuit.
  DoNotAge LSA allowed.
  MD5 authentication SPI 940, secure socket UP (errors: 0)
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/2/4, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 10
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.0.1 (Hello suppressed)
  Suppress hello for 1 neighbor(s)
Ethernet1/0 is up, line protocol is up
  Link Local Address FE80::A8BB:CCFF:FE00:6601, Interface ID 6
  Area 0, Process ID 1, Instance ID 0, Router ID 10.0.0.1
  Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 10.0.0.1, local address FE80::A8BB:CCFF:FE00:6601
  No backup designated router on this network
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
Serial12/0 is up, line protocol is up
  Link Local Address FE80::A8BB:CCFF:FE00:6600, Interface ID 50
  Area 1, Process ID 1, Instance ID 0, Router ID 10.0.0.1
  Network Type POINT_TO_POINT, Cost: 64
  AES-CBC encryption SHA-1 auth SPI 2503, secure socket UP (errors: 0)
  authentication NULL
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Index 1/2/3, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)

```

```

Last flood scan length is 1, maximum is 5
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 10.2.0.1
Suppress hello for 0 neighbor(s)
Serial11/0 is up, line protocol is up
Link Local Address FE80::A8BB:CCFF:FE00:6600, Interface ID 46
Area 1, Process ID 1, Instance ID 0, Router ID 10.0.0.1
Network Type POINT_TO_POINT, Cost: 64
MD5 authentication (Area) SPI 500, secure socket UP (errors: 0)
Transmit Delay is 1 sec, State POINT_TO_POINT,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  Hello due in 00:00:09
Index 1/1/2, flood queue length 0
Next 0x0(0)/0x0(0)/0x0(0)
Last flood scan length is 1, maximum is 5
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 1.0.0.1
Suppress hello for 0 neighbor(s)

```

Sample Output from the show ipv6 ospf Command

The following is sample output from the **show ipv6 ospf** command:

```

Router# show ipv6 ospf

Routing Process "ospfv3 1" with ID 172.16.3.3
It is an autonomous system boundary router
Redistributing External Routes from,
  static
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 240 secs
Interface flood pacing timer 33 msecs
Retransmission pacing timer 66 msecs
Number of external LSA 1. Checksum Sum 0x218D
Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Area 1
    Number of interfaces in this area is 2
    SPF algorithm executed 9 times
    Number of LSA 15. Checksum Sum 0x67581
    Number of DCbitless LSA 0
    Number of indication LSA 0
    Number of DoNotAge LSA 0
    Flood list length 0

```

Sample Output from the show crypto ipsec policy Command

The following is sample output from the **show crypto ipsec policy** command:

```

Router# show crypto ipsec policy

Crypto IPsec client security policy data

Policy name:      OSPFv3-1-1000
Policy refcount:  1
Inbound  AH SPI:  1000 (0x3E8)
Outbound AH SPI:  1000 (0x3E8)
Inbound  AH Key:  1234567890ABCDEF1234567890ABCDEF
Outbound AH Key:  1234567890ABCDEF1234567890ABCDEF
Transform set:    ah-md5-hmac

```

Sample Output from the show crypto ipsec sa ipv6 Command

The following is sample output from the **show crypto ipsec sa ipv6** command:

Router# **show crypto ipsec sa ipv6**

IPv6 IPsec SA info for interface Ethernet0/0

```
protected policy name:OSPFv3-1-1000
IPsec created ACL name:Ethernet0/0-ipsecv6-ACL

local ident (addr/prefixlen/proto/port):(FE80::/10/89/0)
remote ident (addr/prefixlen/proto/port):(::/0/89/0)
current_peer::
  PERMIT, flags={origin_is_acl,}
  #pkts encaps:21, #pkts encrypt:0, #pkts digest:21
  #pkts decaps:20, #pkts decrypt:0, #pkts verify:20
  #pkts compressed:0, #pkts decompressed:0
  #pkts not compressed:0, #pkts compr. failed:0
  #pkts not decompressed:0, #pkts decompress failed:0
  #send errors 0, #recv errors 0

local crypto endpt. ::, remote crypto endpt. ::
path mtu 1500, media mtu 1500
current outbound spi:0x3E8(1000)

inbound ESP SAs:

inbound AH SAs:
  spi:0x3E8(1000)
  transform:ah-md5-hmac ,
  in use settings ={Transport, }
  slot:0, conn_id:2000, flow_id:1, crypto map:N/R
  no sa timing (manual-keyed)
  replay detection support:N

inbound PCP SAs:

outbound ESP SAs:

outbound AH SAs:
  spi:0x3E8(1000)
  transform:ah-md5-hmac ,
  in use settings ={Transport, }
  slot:0, conn_id:2001, flow_id:2, crypto map:N/R
  no sa timing (manual-keyed)
  replay detection support:N

outbound PCP SAs:
```

Sample Output from the show ipv6 ospf graceful-restart Command

The following is sample output from the **show crypto ipsec sa ipv6** command:

Router# **show ipv6 ospf graceful-restart**

```
Routing Process "ospf 1"
Graceful Restart enabled
  restart-interval limit: 120 sec, last restart 00:00:15 ago (took 36 secs)
Graceful Restart helper support enabled
Router status : Active
Router is running in SSO mode
OSPF restart state : NO_RESTART
Router ID 10.1.1.1, checkpoint Router ID 10.0.0.0
```


Configuration Examples for Implementing OSPF for IPv6

This section provides the following configuration examples:

- [Enabling OSPF for IPv6 on an Interface Configuration: Example, page 29](#)
- [Defining an OSPF for IPv6 Area Range: Example, page 29](#)
- [Defining Authentication on an Interface: Example, page 29](#)
- [Defining Authentication in an OSPF Area: Example, page 30](#)
- [Configuring NBMA Interfaces Configuration: Example, page 30](#)
- [Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence: Example, page 30](#)
- [Forcing SPF Configuration: Example, page 30](#)

Enabling OSPF for IPv6 on an Interface Configuration: Example

The following example configures an OSPF routing process 109 to run on the interface and puts it in area 1:

```
ipv6 ospf 109 area 1
```

Defining an OSPF for IPv6 Area Range: Example

The following example specifies an OSPF for IPv6 area range:

```
interface Ethernet7/0
  ipv6 address 2001:0DB8:0:0:7::/64 eui-64
  ipv6 enable
  ipv6 ospf 1 area 1
!
interface Ethernet8/0
  ipv6 address 2001:0DB8:0:0:8::/64 eui-64
  ipv6 enable
  ipv6 ospf 1 area 1
!
interface Ethernet9/0
  ipv6 address 2001:0DB8:0:0:9::/64 eui-64
  ipv6 enable
  ipv6 ospf 1 area 1
!
ipv6 router ospf 1
  router-id 10.11.11.1
  area 1 range 2001:0DB8::/48
```

Defining Authentication on an Interface: Example

The following example defines authentication on the Ethernet 0/0 interface:

```
interface Ethernet0/0
  ipv6 enable
  ipv6 ospf 1 area 0
  ipv6 ospf authentication ipsec spi 500 md5 1234567890ABCDEF1234567890ABCDEF

interface Ethernet0/0
  ipv6 enable
```

```
ipv6 ospf authentication null
ipv6 ospf 1 area 0
```

Defining Authentication in an OSPF Area: Example

The following example defines authentication on OSPF area 0:

```
ipv6 router ospf 1
router-id 11.11.11.1
area 0 authentication ipsec spi 1000 md5 1234567890ABCDEF1234567890ABCDEF
```

Configuring NBMA Interfaces Configuration: Example

The following example configures an OSPF neighboring router with the IPv6 address of FE80::A8BB:CCFF:FE00:C01.

```
interface serial 0
ipv6 enable
ipv6 ospf 1 area 0
encapsulation frame-relay
frame-relay map ipv6 FE80::A8BB:CCFF:FE00:C01 120
ipv6 ospf neighbor FE80::A8BB:CCFF:FE00:C0
```

Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence: Example

The following example displays the configuration values for SPF and LSA throttling timers:

```
Router# show ipv6 ospf

Routing Process "ospfv3 1" with ID 9.9.4.1
Event-log enabled, Maximum number of events: 1000, Mode: cyclic
It is an autonomous system boundary router
Redistributing External Routes from,
  ospf 2
Initial SPF schedule delay 5000 msec
Minimum hold time between two consecutive SPF's 10000 msec
Maximum wait time between two consecutive SPF's 10000 msec
Minimum LSA interval 5 sec
Minimum LSA arrival 1000 msec
```

Forcing SPF Configuration: Example

The following example triggers SPF to redo the SPF and repopulate the routing tables:

```
clear ipv6 ospf force-spf
```

Additional References

The following sections provide additional references related to the Implementing OSPF for IPv6 feature.

Related Documents

Related Topic	Document Title
Configuring a router ID in OSPF	<ul style="list-style-type: none"> • “Configuring OSPF,” Cisco IOS IP Routing Protocols Configuration Guide • Cisco IOS IP Routing Protocols Command Reference
OSPF for IPv6 commands	Cisco IOS IPv6 Command Reference
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
Implementing basic IPv6 connectivity	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPsec for IPv6	“Implementing IPsec for IPv6 Security,” Cisco IOS IPv6 Configuration Guide
Stateful switchover	“Stateful Switchover,” Cisco IOS High Availability Configuration Guide
Cisco nonstop forwarding	“Cisco Nonstop Forwarding,” Cisco IOS High Availability Configuration Guide
OSPF for IPv4 tasks	“Configuring OSPF,” Cisco IOS IP Routing Protocols Configuration Guide
OSPF for IPv4 commands	Cisco IOS IP Routing Protocols Command Reference
Security configuration tasks (IPv4)	Cisco IOS Security Configuration Guide
Security commands: complete command syntax, command mode, defaults, usage guidelines, and examples (IPv4)	Cisco IOS Security Command Reference
LSA throttling	“OSPF Link-State Advertisement (LSA) Throttling,” Cisco IOS IP Routing Protocols Configuration Guide

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> CISCO-IETF-IP-FORWARD-MIB CISCO-IETF-IP-MIB 	<p>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
RFC 2401	Security Architecture for the Internet Protocol
RFC 2402	<i>IP Authentication Header</i>
RFC 2406	<i>IP Encapsulating Security Payload (ESP)</i>
RFC 2740	<i>OSPF for IPv6</i>
RFC 4552	<i>Authentication/Confidentiality for OSPFv3</i>

The draft RFC supported is as follows:

- draft-ietf-ospf-ospfv3-graceful-restart, *OSPFv3 Graceful Restart*

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about

all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **area authentication (IPv6)**
- **area encryption**
- **area range**
- **area virtual-link**
- **area virtual-link authentication**
- **area virtual-link encryption**
- **auto-cost (IPv6)**
- **clear ipv6 ospf**
- **clear ipv6 ospf counters**
- **clear ipv6 ospf events**
- **debug ipv6 ospf**
- **debug ipv6 ospf database-timer rate-limit**
- **debug ipv6 ospf events**
- **debug ipv6 ospf lsdb**
- **debug ipv6 ospf monitor**
- **debug ipv6 ospf packet**
- **debug ipv6 ospf spf statistic**
- **default (IPv6 OSPF)**
- **default-information originate (IPv6 OSPF)**
- **default-metric (IPv6)**
- **discard-route (IPv6)**
- **distribute-list prefix-list (IPv6 OSPF)**
- **event-log**
- **graceful-restart**
- **graceful-restart helper**
- **ipv6 ospf area**
- **ipv6 ospf authentication**
- **ipv6 ospf cost**
- **ipv6 ospf database-filter all out**
- **ipv6 ospf dead-interval**
- **ipv6 ospf demand-circuit**
- **ipv6 ospf encryption**
- **ipv6 ospf flood-reduction**
- **ipv6 ospf hello-interval**
- **ipv6 ospf mtu-ignore**

- **ipv6 ospf name-lookup**
- **ipv6 ospf neighbor**
- **ipv6 ospf network**
- **ipv6 ospf priority**
- **ipv6 ospf retransmit-interval**
- **ipv6 ospf transmit-delay**
- **ipv6 router ospf**
- **log-adjacency-changes**
- **passive-interface (IPv6)**
- **process-min-time percent**
- **router-id (IPv6)**
- **show ipv6 ospf**
- **show ipv6 ospf border-routers**
- **show ipv6 ospf database**
- **show ipv6 ospf event**
- **show ipv6 ospf flood-list**
- **show ipv6 ospf graceful-restart**
- **show ipv6 ospf interface**
- **show ipv6 ospf neighbor**
- **show ipv6 ospf request-list**
- **show ipv6 ospf retransmission-list**
- **show ipv6 ospf summary-prefix**
- **show ipv6 ospf timers rate-limit**
- **show ipv6 ospf virtual-links**
- **summary-prefix (IPv6 OSPF)**
- **timers lsa arrival**
- **timers pacing flood (IPv6)**
- **timers pacing lsa-group (IPv6)**
- **timers pacing retransmission (IPv6)**
- **timers spf (IPv6)**
- **timers throttle lsa**
- **timers throttle spf**

Feature Information for Implementing OSPF for IPv6

Table 3 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.0(24)S or a later release appear in the table.

For information on a feature in this technology that is not documented here, see [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#).

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 3 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 3 Feature Information for Implementing OSPF for IPv6

Feature Name	Releases	Feature Information
IPv6 routing: OSPF for IPv6 (OSPFv3)	12.0(24)S 12.2(18)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	OSPF version 3 for IPv6 expands on OSPF version 2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses. This entire document provides information about this feature.
IPv6 routing: LSA types in OSPF for IPv6	12.0(24)S 12.2(18)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	A router's collection of LSA data is stored in a link-state database. The contents of the database, when subjected to the Dijkstra algorithm, result in the creation of the OSPF routing table. The following sections provide information about this feature: <ul style="list-style-type: none"> • How OSPF for IPv6 Works, page 3 • LSA Types for IPv6, page 3

Table 3 **Feature Information for Implementing OSPF for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 routing: Fast Convergence—LSA and SPF throttling	12.2(33)SB 12.2(33)SRC Cisco IOS XE Release 2.1	<p>The OSPF for IPv6 LSA and SPF throttling feature provides a dynamic mechanism to slow down link-state advertisement updates in OSPF during times of network instability.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Fast Convergence—LSA and SPF Throttling, page 5 • Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence, page 19 • Enabling Event Logging for LSA and SPF Rate Limiting, page 20 • Clearing the Content of an Event Log, page 21 • Configuring LSA and SPF Throttling for OSPF for IPv6 Fast Convergence: Example, page 30
IPv6 routing: NBMA interfaces in OSPF for IPv6	12.0(24)S 12.2(18)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>On NBMA networks, the DR or backup DR performs the LSA flooding.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • NBMA in OSPF for IPv6, page 5 • Configuring NBMA Interfaces, page 18
IPv6 routing: Force SPF in OSPF for IPv6	12.0(24)S 12.2(18)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>This feature enables the OSPF database to be cleared and repopulated, and then the SPF algorithm is performed.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Force SPF in OSPF for IPv6, page 5 • Enabling OSPFv3 Graceful Restart, page 22
IPv6 routing: Load balancing in OSPF for IPv6	12.0(24)S 12.2(18)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>OSPF for IPv6 performs load balancing automatically.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Load Balancing in OSPF for IPv6, page 5

Table 3 *Feature Information for Implementing OSPF for IPv6 (continued)*

Feature Name	Releases	Feature Information
IPv6 routing: OSPF for IPv6 authentication support with IPsec	12.3(4)T 12.4 12.4(2)T	<p>OSPF for IPv6 uses the IPsec secure socket API to add authentication to OSPF for IPv6 packets.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • OSPF for IPv6 Authentication Support with IPsec, page 6 • Configuring IPsec on OSPF for IPv6, page 13 • Defining Authentication on an Interface, page 13 • Defining Authentication in an OSPF Area, page 15
IPv6 routing: OSPF IPv6 (OSPFv3) IPsec ESP encryption and authentication	12.4(9)T	<p>IPv6 ESP extension headers can be used to provide authentication and confidentiality to OSPF for IPv6.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing OSPF for IPv6, page 2 • OSPF for IPv6 Authentication Support with IPsec, page 6 • Defining Encryption on an Interface, page 14 • Defining Encryption in an OSPF Area, page 16 • Defining Authentication and Encryption for a Virtual Link in an OSPF Area, page 17
OSPFv3 dynamic interface cost support	12.4(15)T	<p>OSPFv3 dynamic interface cost support provides enhancements to the OSPF for IPv6 cost metric for supporting mobile ad hoc networking.</p> <p>The following section provides information about this feature:</p> <ul style="list-style-type: none"> • OSPF Cost Calculation, page 8
OSPFv3 graceful restart	Cisco IOS XE Release 2.1	<p>The graceful restart feature in OSPFv3 allows nonstop data forwarding along routes that are already known while the OSPFv3 routing protocol information is being restored.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • OSPFv3 Graceful Restart, page 10 • Enabling OSPFv3 Graceful Restart, page 22

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 over MPLS

First Published: March 17, 2003

Last Updated: May 1, 2006

Multiprotocol Label Switching (MPLS) is deployed by many service providers in their IPv4 networks. Service providers want to introduce IPv6 services to their customers, but changes to their existing IPv4 infrastructure can be expensive and the cost benefit for a small amount of IPv6 traffic does not make economic sense. Several integration scenarios have been developed to leverage an existing IPv4 MPLS infrastructure and add IPv6 services without requiring any changes to the network backbone. This document describes how to implement IPv6 over MPLS.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPv6 over MPLS” section on page 18](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IPv6 over MPLS, page 2](#)
- [Information About Implementing IPv6 over MPLS, page 2](#)
- [How to Implement IPv6 over MPLS, page 5](#)
- [Configuration Examples for IPv6 over MPLS, page 13](#)
- [Where to Go Next, page 15](#)
- [Additional References, page 16](#)
- [Command Reference, page 17](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2003–2009 Cisco Systems, Inc. All rights reserved.

- [Feature Information for Implementing IPv6 over MPLS, page 18](#)

Prerequisites for Implementing IPv6 over MPLS

- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the [“Related Documents”](#) section for IPv4 configuration and command reference information.
- Before the IPv6 Provider Edge Router over MPLS (6PE) feature can be implemented, MPLS must be running over the core IPv4 network. If Cisco routers are used, Cisco Express Forwarding or distributed Cisco Express Forwarding must be enabled for both IPv4 and IPv6 protocols. This module assumes that you are familiar with MPLS.

Information About Implementing IPv6 over MPLS

To configure IPv6 over MPLS, you need to understand the following concepts:

- [Benefits of Deploying IPv6 over MPLS Backbones, page 2](#)
- [IPv6 over a Circuit Transport over MPLS, page 2](#)
- [IPv6 Using Tunnels on the Customer Edge Routers, page 3](#)
- [IPv6 on the Provider Edge Routers \(6PE\), page 4](#)

Benefits of Deploying IPv6 over MPLS Backbones

IPv6 over MPLS backbones enables isolated IPv6 domains to communicate with each other over an MPLS IPv4 core network. This implementation requires only a few backbone infrastructure upgrades and no reconfiguration of core routers because forwarding is based on labels rather than the IP header itself, providing a very cost-effective strategy for the deployment of IPv6.

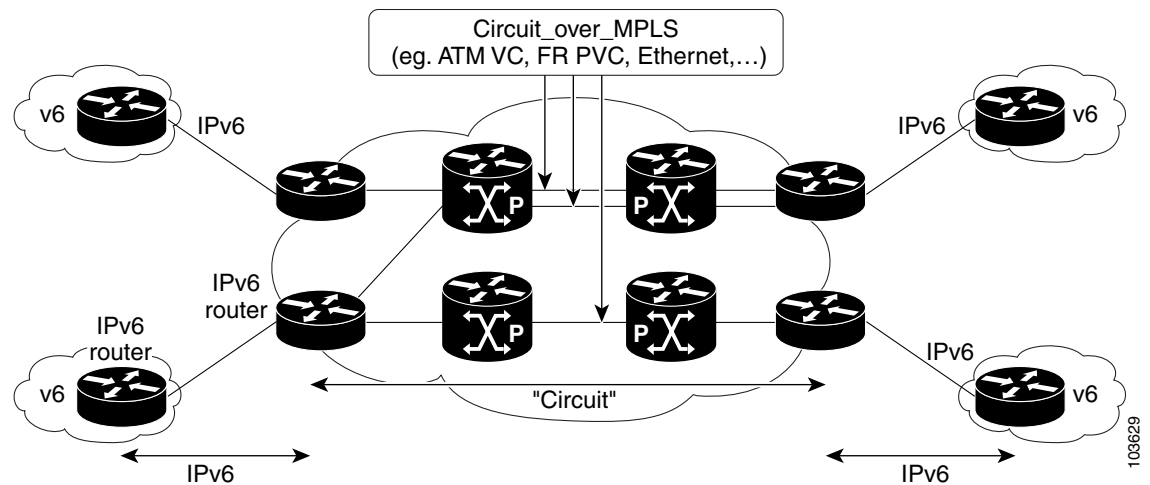
Additionally, the inherent Virtual Private Network (VPN) and MPLS traffic engineering (MPLS-TE) services available within an MPLS environment allow IPv6 networks to be combined into IPv4 VPNs or extranets over an infrastructure supporting IPv4 VPNs and MPLS-TE.

IPv6 over a Circuit Transport over MPLS

Using any circuit transport for deploying IPv6 over MPLS networks has no impact on the operation or infrastructure of MPLS, and requires no configuration changes to the core or provider edge routers. Communication between the remote IPv6 domains runs native IPv6 protocols over a dedicated link, where the underlying mechanisms are fully transparent to IPv6. The IPv6 traffic is tunneled using the Any Transport over MPLS (MPLS/AToM) or Ethernet over MPLS (EoMPLS) feature with the routers connected through an ATM OC-3 or Ethernet interface, respectively.

[Figure 21](#) shows the configuration for IPv6 over any circuit transport over MPLS.

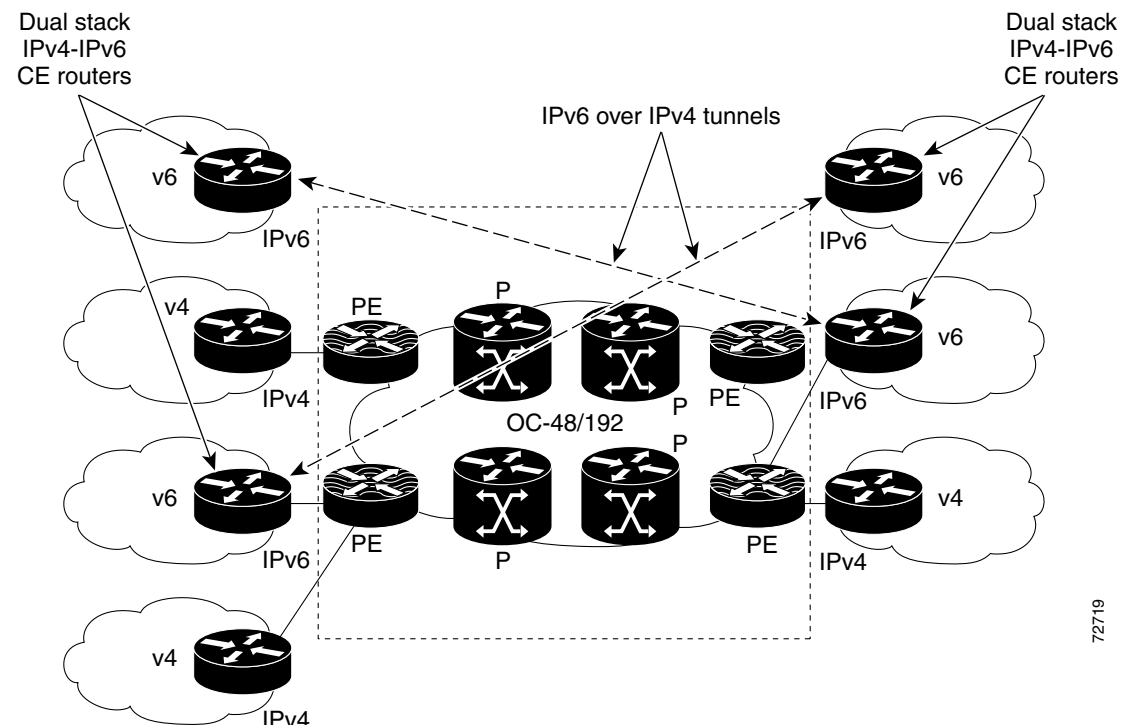
Figure 21 IPv6 over a Circuit Transport over MPLS



IPv6 Using Tunnels on the Customer Edge Routers

Using tunnels on the customer edge (CE) routers is the simplest way of deploying IPv6 over MPLS networks with no impact on the operation or infrastructure of MPLS, and no configuration changes to the core or provider edge routers. Communication between the remote IPv6 domains uses standard tunneling mechanisms and requires the CE routers to be configured to run dual IPv4 and IPv6 protocol stacks. [Figure 22](#) shows the configuration using tunnels on the CE routers.

Figure 22 IPv6 Using Tunnels on the CE Routers



Refer to [Implementing Tunneling for IPv6](#) for configuration information on manually configured tunnels, automatic tunnels, and 6to4 tunnels.

Limitations on using tunnels involve the manual configuring of a mesh of tunnels on the CE routers, creating scaling issues for large networks.

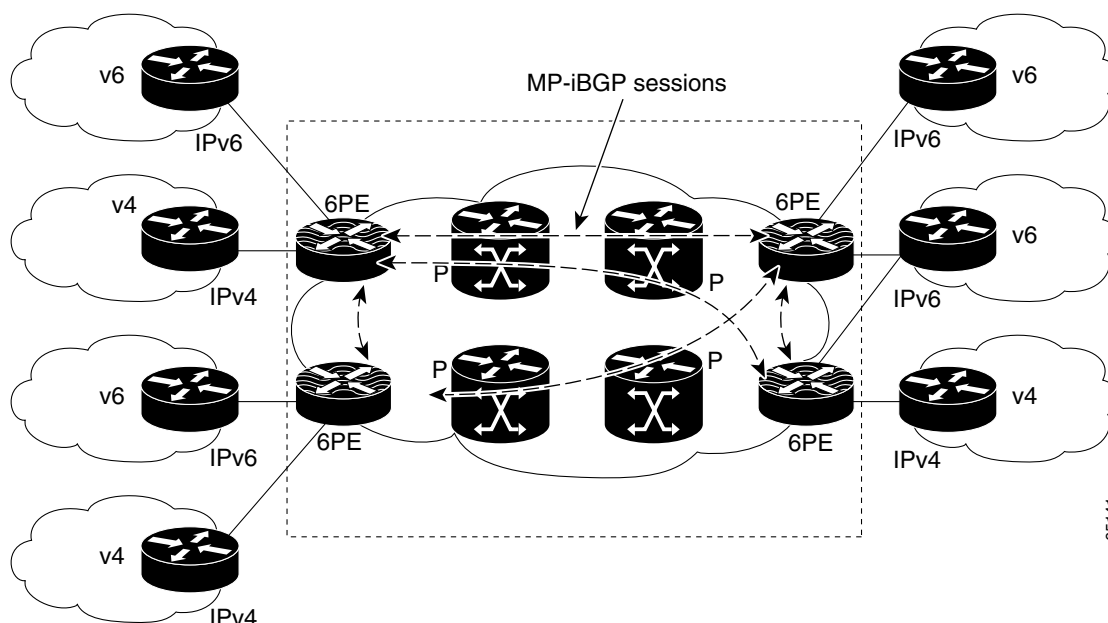
IPv6 on the Provider Edge Routers (6PE)

The Cisco implementation of IPv6 provider edge router over MPLS is called 6PE, and it enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS label switched paths (LSPs). This feature relies on multiprotocol Border Gateway Protocol (BGP) extensions in the IPv4 network configuration on the provider edge (PE) router to exchange IPv6 reachability information in addition to an MPLS label for each IPv6 address prefix to be advertised. Edge routers are configured to be dual stack running both IPv4 and IPv6, and use the IPv4 mapped IPv6 address for IPv6 prefix reachability exchange.

A hierarchy of labels is imposed on the 6PE ingress router to keep the IPv6 traffic transparent to all the core routers. The top label provides connectivity inside the IPv4 MPLS core network and the label is distributed by Label Distribution Protocol (LDP), Tag Distribution Protocol (TDP), or Resource Reservation Protocol (RSVP). TDP and LDP can both be used for label distribution, but RSVP is used only in the context of MPLS-TE label exchange. The bottom label, automatically assigned to the IPv6 prefix of the destination, is distributed by multiprotocol BGP and used at each 6PE egress router for IPv6 forwarding.

In [Figure 23](#) the 6PE routers are configured as dual stack routers able to route both IPv4 and IPv6 traffic. Each 6PE router is configured to run LDP, TDP, or RSVP (if traffic engineering is configured) to bind the IPv4 labels. The 6PE routers use multiprotocol BGP to exchange reachability information with the other 6PE devices within the MPLS domain, and to distribute aggregate IPv6 labels between them. All 6PE and core routers—P routers in [Figure 3](#)—within the MPLS domain share a common IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Integrated Intermediate System-to-Intermediate System (IS-IS).

Figure 23 6PE Router Topology



65141

The interfaces on the 6PE routers connecting to the CE router can be configured to forward IPv6 traffic, IPv4 traffic, or both types of traffic depending on the customer requirements. 6PE routers advertise IPv6 reachability information learned from their 6PE peers over the MPLS cloud. Service providers can delegate an IPv6 prefix from their registered IPv6 prefixes over the 6PE infrastructure; otherwise, there is no impact on the CE router.

The P routers in the core of the network are not aware that they are switching IPv6 packets. Core routers are configured to support MPLS and the same IPv4 IGP as the PE routers to establish internal reachability inside the MPLS cloud. Core routers also use LDP, TDP, or RSVP for binding IPv4 labels. Implementing the Cisco 6PE feature does not have any impact on the MPLS core devices.

Within the MPLS network, IPv6 traffic is forwarded using label switching, making the IPv6 traffic transparent to the core of the MPLS network. No IPv6 over IPv4 tunnels or Layer 2 encapsulation methods are required.

6PE Multipath

Internal and external Border Gateway Protocol (BGP) multipath for IPv6 allows the IPv6 router to load balance between several paths (for example, same neighboring autonomous system [AS] or sub-AS, or the same metric) to reach its destination. The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.

When MP-iBGP multipath is enabled on the 6PE router, all labeled paths are installed in the forwarding table with MPLS information (label stack) when MPLS information is available. This functionality enables 6PE to perform load balancing.

How to Implement IPv6 over MPLS

The following sections explain how to configure IPv6 over MPLS:

- [Deploying IPv6 over a Circuit Transport over MPLS, page 5](#)
- [Deploying IPv6 on the Provider Edge Routers \(6PE\), page 5](#)
- [Configuring iBGP Multipath Load Sharing, page 9](#)
- [Verifying 6PE Configuration and Operation, page 10](#)

Deploying IPv6 over a Circuit Transport over MPLS

To deploy IPv6 over a circuit transport over MPLS, the IPv6 routers must be configured for IPv6 connectivity. Refer to [Implementing IPv6 Addressing and Basic Connectivity](#) for details on basic IPv6 configuration. The MPLS router configuration requires AToM configuration or EoMPLS configuration.

Deploying IPv6 on the Provider Edge Routers (6PE)

To implement IPv6 on provider edge routers two tasks must be completed. The first task is to specify the interface from which locally generated packets take their source IPv6 address. The second task is to bind and advertise aggregate labels.

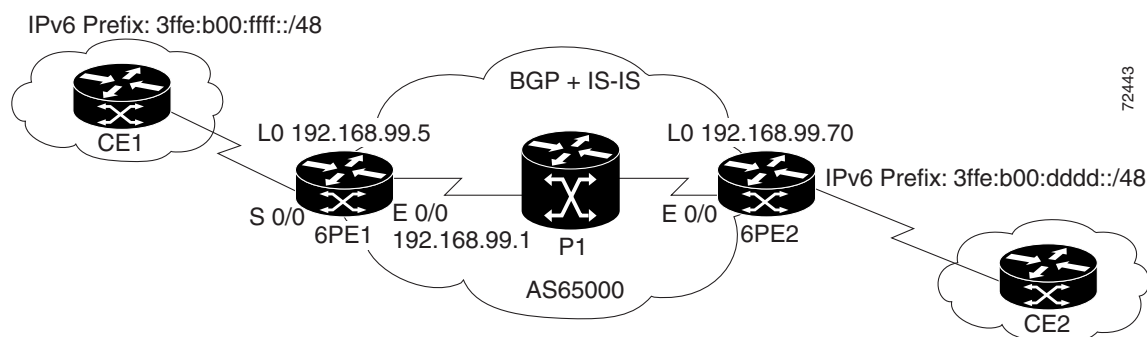
Each 6PE router—6PE1 and 6PE2 in [Figure 24](#)—is assumed to be running IPv4 routing and Cisco Express Forwarding.

6PE Network Configuration

Two configuration tasks using the network shown in [Figure 24](#) are required at the 6PE1 router to enable the 6PE feature.

The customer edge router—CE1 in [Figure 24](#)—is configured to forward its IPv6 traffic to the 6PE1 router. The P1 router in the core of the network is assumed to be running MPLS, a label distribution protocol, an IPv4 IGP, and Cisco Express Forwarding or distributed Cisco Express Forwarding, and does not require any new configuration to enable the 6PE feature. Although new configuration tasks are not required for the CE1 and P1 routers, configuration examples are shown for reference in the [“Configuration Examples for IPv6 over MPLS”](#) section on [page 13](#).

Figure 24 6PE Configuration Example



Prerequisites

- The 6PE routers—the 6PE1 and 6PE2 routers in [Figure 24](#)—must be members of the core IPv4 network. The 6PE router interfaces attached to the core network must be running MPLS, the same label distribution protocol, and the same IPv4 IGP, as in the core network.
- The 6PE routers must also be configured to be dual stack to run both IPv4 and IPv6.

Restrictions



Note

As of Cisco IOS Release 12.2(22)S, the following restrictions do not apply to Cisco IOS 12.2 S releases.

The following restrictions apply when implementing the IPv6 Provider Edge Router over MPLS (6PE) feature:

- Core MPLS routers are supporting MPLS and IPv4 only, so they cannot forward or create any IPv6 Internet Control Message Protocol (ICMP) messages.
- Load balancing ability is not provided by Cisco 6PE between an MPLS path and an IPv6 path. If both are available, the MPLS path is always preferred. Load balancing between two MPLS paths is possible.
- BGP multipath is not supported for Cisco 6PE routes. If two BGP peers advertise the same prefix with an equal cost, Cisco 6PE will use the last route to cross the MPLS core.
- 6PE feature is not supported over tunnels other than RSVP-TE tunnels.

Specifying the Source Address Interface on a 6PE Router

This task explains how to specify the interface from which locally generated packets take their source IPv6 address. This task is the first of two tasks that must be completed to deploy 6PE. See the [“Binding and Advertising the 6PE Label to Advertise Prefixes”](#) task for details about the second task required to implement 6PE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **ipv6 cef**
5. **interface** *type number*
6. **ipv6 address** {*ipv6-address/prefix-length* | *prefix-name sub-bits/prefix-length*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	ipv6 cef Example: Router(config)# ipv6 cef	Enables IPv6 Cisco Express Forwarding.
Step 5	interface <i>type number</i> Example: Router(config)# interface Serial 0/0	Specifies an interface type and number and enters interface configuration mode. <ul style="list-style-type: none">In the context of this feature, the interface to be configured is the interface communicating with the CE router.
Step 6	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if)# ipv6 address 2001:0DB8:FFFF::2/64	Configures an IPv6 address based on an IPv6 general prefix and enable IPv6 processing on an interface.

Binding and Advertising the 6PE Label to Advertise Prefixes

This task explains how to enable the binding and advertising of aggregate labels when advertising IPv6 prefixes to a specified BGP neighbor.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **no bgp default ipv4-unicast**
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
7. **address-family ipv6** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	no bgp default ipv4-unicast Example: Router(config-router)# no bgp default ipv4-unicast	Disables the IPv4 unicast address family for the BGP routing process specified in the previous step. Note Routing information for the IPv4 unicast address family is advertised by default for each BGP routing session configured with the neighbor remote-as command unless you configure the no bgp default ipv4-unicast command before configuring the neighbor remote-as command.

	Command or Action	Purpose
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.99.70 remote-as 65000	Adds the IP address of the neighbor in the specified autonomous system to the BGP neighbor table of the local router.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.99.70 update-source Loopback 0	Specifies the interface whose IPv4 address is to be used as the source address for the peering. <ul style="list-style-type: none"> In the context of this task, the interface must have an IPv4 address with a 32-bit mask configured. Use of a loopback interface is recommended. This address is used to determine the IPv6 next hop by the peer 6PE.
Step 7	address-family ipv6 [unicast] Example: Router(config-router)# address-family ipv6	Specifies the IPv6 address family and enters address family configuration mode. <ul style="list-style-type: none"> The unicast keyword specifies the IPv6 unicast address family. By default, the router is placed in configuration mode for the IPv6 unicast address family if the unicast keyword is not specified with the address-family ipv6 command.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.99.70 activate	Enables the neighbor to exchange prefixes for the IPv6 address family with the local router.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> } send-label Example: Router(config-router-af)# neighbor 192.168.99.70 send-label	Advertises the capability of the router to send MPLS labels with BGP routes. <ul style="list-style-type: none"> In IPv6 address family configuration mode this command enables binding and advertisement of aggregate labels when advertising IPv6 prefixes in BGP.

Configuring iBGP Multipath Load Sharing

This task describes how to configure iBGP multipath load sharing and control the maximum number of parallel iBGP routes that can be installed in a routing table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *as-number*
4. **maximum-paths ibgp** *number-of-paths*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>as-number</i> Example: Router(config)# router bgp 65000	Enters router configuration mode for the specified routing process.
Step 4	maximum-paths ibgp <i>number-of-paths</i> Example: Router(config-router)# maximum-paths ibgp 3	Controls the maximum number of parallel iBGP routes that can be installed in a routing table.

Verifying 6PE Configuration and Operation

When 6PE is running the following components can be monitored:

- Multiprotocol BGP
- MPLS
- Cisco Express Forwarding for IPv6
- IPv6 routing table

This optional task explains how to display information about the various components to verify the configuration and operation of 6PE.

SUMMARY STEPS

1. **show bgp ipv6** {unicast | multicast} [*ipv6-prefix/prefix-length*] [**longer-prefixes**] [**labels**]
2. **show bgp ipv6** {unicast | multicast} **neighbors** [*ipv6-address*] [**received-routes** | **routes** | **flap-statistics** | **advertised-routes** | **paths** *regular-expression* | **dampened-routes**]
3. **show mpls forwarding-table** [*network {mask | length}*] | **labels** *label* [-*label*] | **interface** *interface* | **next-hop** *address* | **lsp-tunnel** [*tunnel-id*] [**vrf** *vrf-name*] [**detail**]
4. **show ipv6 cef** [*ipv6-prefix/prefix-length*] | [*interface-type interface-number*] [**longer-prefixes** | **similar-prefixes** | **detail** | **internal** | **platform** | **epoch** | **source**]
5. **show ipv6 route** [*ipv6-address* | *ipv6-prefix/prefix-length* | *protocol* | *interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	show bgp ipv6 {unicast multicast} <i>[ipv6-prefix/prefix-length] [longer-prefixes]</i> <i>[labels]</i> Example: Router> show bgp ipv6 unicast 2001:0DB8:DDDD::/48	(Optional) Displays entries in the IPv6 BGP routing table. <ul style="list-style-type: none"> In this example, information about the IPv6 route for the prefix 2001:0DB8:DDDD::/48 is displayed.
Step 2	show bgp ipv6 {unicast multicast} neighbors <i>[ipv6-address] [received-routes routes </i> <i>flap-statistics advertised-routes paths</i> <i>regular-expression dampened-routes]</i> Example: Router> show bgp ipv6 neighbors unicast 192.168.99.70	(Optional) Displays information about IPv6 BGP connections to neighbors. <ul style="list-style-type: none"> In this example, information including the IPv6 label capability is displayed for the BGP peer at 192.168.99.70.
Step 3	show mpls forwarding-table <i>[network {mask </i> <i>length} labels label [-label] interface</i> <i>interface nexthop address lsp-tunnel</i> <i>[tunnel-id]] [vrf vrf-name] [detail]</i> Example: Router> show mpls forwarding-table	(Optional) Displays the contents of the MPLS Forwarding Information Base (FIB). <ul style="list-style-type: none"> In this example, information linking the MPLS label with IPv6 prefixes is displayed where the labels are shown as aggregate and the prefix is shown as IPv6.
Step 4	show ipv6 cef <i>[ipv6-prefix/prefix-length] </i> <i>[interface-type interface-number]</i> <i>[longer-prefixes similar-prefixes detail </i> <i>internal platform epoch source]]</i> Example: Router> show ipv6 cef 2001:0DB8:DDDD::/64	(Optional) Displays FIB entries based on IPv6 address information. <ul style="list-style-type: none"> In this example, label information from the Cisco Express Forwarding table for prefix 2001:0DB8:DDDD::/64 is displayed.
Step 5	show ipv6 route <i>[ipv6-address </i> <i>ipv6-prefix/prefix-length protocol </i> <i>interface-type interface-number]</i> Example: Router> show ipv6 route	(Optional) Displays the current contents of the IPv6 routing table.

Output Examples

This section provides the following output examples:

- [Sample Output from the show bgp ipv6 Command, page 12](#)
- [Sample Output from the show bgp ipv6 neighbors Command, page 12](#)
- [Sample Output from the show mpls forwarding-table Command, page 12](#)
- [Sample Output from the show bgp ipv6 Command, page 13](#)
- [Sample Output from the show ipv6 cef Command, page 13](#)
- [Sample Output from the show ipv6 route Command, page 13](#)

Sample Output from the show bgp ipv6 Command

In the following example, output information about an IPv6 route is displayed using the **show bgp ipv6** command with an IPv6 prefix:

```
Router# show bgp ipv6 2001:0DB8:DDDD::/48

BGP routing table entry for 2001:0DB8:DDDD::/48, version 15
Paths: (1 available, best #1, table Global-IPv6-Table)
  Not advertised to any peer
  Local
    ::FFFF:192.168.99.70 (metric 20) from 192.168.99.70 (192.168.99.70)
      Origin IGP, localpref 100, valid, internal, best
```

Sample Output from the show bgp ipv6 neighbors Command

In the following example, output information about a BGP peer including the “IPv6 label” capability is displayed using the **show bgp ipv6 neighbors** command with an IP address:

```
Router# show bgp ipv6 neighbors 192.168.99.70

BGP neighbor is 192.168.99.70, remote AS 65000, internal link
  BGP version 4, remote router ID 192.168.99.70
  BGP state = Established, up for 00:05:17
  Last read 00:00:09, hold time is 0, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv6 Unicast: advertised and received
    ipv6 MPLS Label capability: advertised and received
  Received 54 messages, 0 notifications, 0 in queue
  Sent 55 messages, 1 notifications, 0 in queue
  Default minimum time between advertisement runs is 5 seconds

For address family: IPv6 Unicast
  BGP table version 21880, neighbor version 21880
  Index 1, Offset 0, Mask 0x2
  Route refresh request: received 0, sent 0
  77 accepted prefixes consume 4928 bytes
  Prefix advertised 4303, suppressed 0, withdrawn 1328
  Number of NLRI in the update sent: max 1, min 0
```

Sample Output from the show mpls forwarding-table Command

In the following example, output information linking the MPLS label with prefixes is displayed using the **show mpls forwarding-table** command. If the 6PE feature is configured, the labels are aggregated because there are several prefixes for one local label, and the prefix column contains “IPv6” instead of a target prefix.

```
Router# show mpls forwarding-table
```

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes switched	Outgoing interface	Next Hop
16	Aggregate	IPv6	0		
17	Aggregate	IPv6	0		
18	Aggregate	IPv6	0		
19	Pop tag	192.168.99.64/30	0	Se0/0	point2point
20	Pop tag	192.168.99.70/32	0	Se0/0	point2point
21	Pop tag	192.168.99.200/32	0	Se0/0	point2point
22	Aggregate	IPv6	5424		
23	Aggregate	IPv6	3576		
24	Aggregate	IPv6	2600		

Sample Output from the show bgp ipv6 Command

In the following example, output information about the top of the stack label with label switching information is displayed using the **show bgp ipv6** command with the **labels** keyword:

```
Router# show bgp ipv6 labels
```

```
Network                Next Hop                In tag/Out tag
2001:0DB8:DDDD::/64    ::FFFF:192.168.99.70    notag/20
```

Sample Output from the show ipv6 cef Command

In the following example, output information about labels from the Cisco Express Forwarding table is displayed using the **show ipv6 cef** command with an IPv6 prefix:

```
Router# show ipv6 cef 2001:0DB8:DDDD::/64
```

```
2001:0DB8:DDDD::/64
  nexthop ::FFFF:192.168.99.70
  fast tag rewrite with Se0/0, point2point, tags imposed {19 20}
```

Sample Output from the show ipv6 route Command

In the following example, output information from the IPv6 routing table is displayed using the **show ipv6 route** command. The output shows the IPv6 MPLS virtual interface as the output interface of IPv6 routes forwarded across the MPLS cloud. In this example using the routers in [Figure 24](#), the output is from the 6PE1 router.

The 6PE2 router has advertised the IPv6 prefix of 2001:0DB8:dddd::/48 configured for the CE2 router and the next-hop address is the IPv4-compatible IPv6 address ::ffff:192.168.99.70, where 192.168.99.70 is the IPv4 address of the 6PE2 router.

```
Router# show ipv6 route
```

```
IPv6 Routing Table - 10 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
B 2001:0DB8:DDDD::/64 [200/0]
   via ::FFFF:192.168.99.70, IPv6-mpls
B 2001:0DB8:DDDD::/64 [200/0]
   via ::FFFF:192.168.99.70, IPv6-mpls
L 2001:0DB8:FFFF::1/128 [0/0]
   via ::, Ethernet0/0
C 2001:0DB8:FFFF::/64 [0/0]
   via ::, Ethernet0/0
S 2001:0DB8:FFFF::/48 [1/0]
   via 2001:0DB8:B00:FFFF::2, Ethernet0/0
```

Configuration Examples for IPv6 over MPLS

The following examples show 6PE configuration examples for three of the routers shown in [Figure 24](#) and used in the “[Specifying the Source Address Interface on a 6PE Router](#)” and “[Binding and Advertising the 6PE Label to Advertise Prefixes](#)” sections.

- [Customer Edge Router: Example, page 14](#)
- [Provider Edge Router: Example, page 14](#)
- [Core Router: Example, page 15](#)

Customer Edge Router: Example

In the following example, serial interface 0/0 of the customer edge router—CE1 in [Figure 24](#)—is connected to the service provider and is assigned an IPv6 address. IPv6 is enabled and a default static route is installed using the IPv6 address of serial interface 0/0 of the 6PE1 router.

```
ip cef
!
ipv6 unicast-routing
!
interface Serial 0/0
  description to_6PE1_router
  no ip address
  ipv6 address 2001:0DB8:FFFF::2/64
!
ipv6 route ::/0 Serial 0/0 FE80::210:XXXX:FE01:1001
```

Provider Edge Router: Example

The 6PE router—Router 6PE1 in [Figure 24](#)—is configured for both IPv4 and IPv6 traffic. Ethernet interface 0/0 is configured with an IPv4 address and is connected to a router in the core of the network—router P1 in [Figure 24](#). Integrated IS-IS and TDP configurations on this router are similar to the P1 router.

Router 6PE1 exchanges IPv6 routing information with another 6PE router—Router 6PE2 in [Figure 24](#)—using internal BGP (iBGP) established over an IPv4 connection so that all the **neighbor** commands use the IPv4 address of the 6PE2 router. All the BGP peers are within autonomous system 65000, so synchronization with IGP is turned off for IPv4. In IPv6 address family configuration mode, synchronization is disabled by default.

IPv6 and Cisco Express Forwarding for IPv6 are enabled, the 6PE2 neighbor is activated, and aggregate label binding and advertisement is enabled for IPv6 prefixes using the **neighbor send-label** command. Connected and static IPV6 routes are redistributed using BGP. If IPv6 packets are generated in the local router, the IPv6 address for MPLS processing will be the address of loopback interface 0.

In the following example, serial interface 0/0 connects to the customer and the IPv6 prefix delegated to the customer is 2001:0DB8:ffff::/48, which is determined from the service provider IPv6 prefix. A static route is configured to route IPv6 packets between the 6PE route and the CE router.

```
ip cef
ipv6 cef
ipv6 unicast-routing
!
mpls ipv6 source-interface Loopback0
tag-switching tdp router-id Loopback0
!
interface Loopback0
  ip address 192.168.99.5 255.255.255.255
  ipv6 address 2001:0DB8:1000:1::1/64
!
interface Ethernet0/0
  description to_P_router
  ip address 192.168.99.1 255.255.255.252
  ip router isis
  tag-switching ip
!
interface Serial0/0
  description to_CE_router
  no ip address
```



```

    ipv6 address 2001:0DB8:FFFF::1/64
    !
router isis
    passive-interface Loopback0
    net 49.0001.1921.6809.9005.00
    !
router bgp 65000
    no bgp default ipv4-unicast
    bgp log-neighbor-changes
    neighbor 192.168.99.70 remote-as 65000
    neighbor 192.168.99.70 description to_6PE2
    neighbor 192.168.99.70 update-source Loopback0
    !
    address-family ipv6
    neighbor 192.168.99.70 activate
    neighbor 192.168.99.70 send-label
    network 2001:0DB8:FFFF::/48
    exit-address-family
    !
ipv6 route 2001:0DB8:FFFF::/48 Ethernet0/0 2001:0DB8:FFFF::2

```

Core Router: Example

In the following example, the router in the core of the network—Router P in [Figure 24](#)—is running MPLS, IS-IS, and IPv4 only. The Ethernet interfaces are configured with IPv4 address and are connected to the 6PE routers. IS-IS is the IGP for this network and the P1 and 6PE routers are in the same IS-IS area 49.0001. TDP and tag switching are enabled on both the Ethernet interfaces. Cisco Express Forwarding is enabled in global configuration mode.

```

ip cef
!
tag-switching tdp router-id Loopback0
!
interface Loopback0
    ip address 192.168.99.200 255.255.255.255
    !
interface Ethernet0/0
    description to_6PE1
    ip address 192.168.99.2 255.255.255.252
    ip router isis
    tag-switching ip
    !
interface Ethernet0/1
    description to_6PE2
    ip address 192.168.99.66 255.255.255.252
    ip router isis
    tag-switching ip

router isis
    passive-interface Loopback0
    net 49.0001.1921.6809.9200.00

```

Where to Go Next

If you want to further customize your MPLS network, refer to the [Cisco IOS IP Switching Configuration Guide](#).

Additional References

The following sections provide references related to the Implementing IPv6 over MPLS feature.

Related Documents

Related Topic	Document Title
IPv6 using tunnels on the CE routers	“Implementing Tunneling for IPv6,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
MPLS configuration tasks	“Multiprotocol Label Switching Overview,” Cisco IOS Multiprotocol Label Switching Configuration Guide
MPLS commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS Multiprotocol Label Switching Command Reference

Standards

Standards	Title
Draft-ietf-ngtrans-bgp-tunnel-04.txt	Connecting IPv6 Islands Across IPv4 Clouds with BGP

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **address-family ipv6**
- **ipv6 cef**
- **ipv6 unicast-routing**
- **maximum-paths ibgp**
- **mpls ipv6 source-interface**
- **neighbor activate**
- **neighbor remote-as**
- **neighbor send-label**
- **neighbor update-source**
- **router bgp**
- **show bgp ipv6**
- **show bgp ipv6 neighbors**
- **show mpls forwarding-table**
- **show ipv6 cef**
- **show ipv6 route**

Feature Information for Implementing IPv6 over MPLS

Table 8 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(14)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 8 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 8 Feature Information for Implementing IPv6 over MPLS

Feature Name	Releases	Feature Information
IPv6 over a circuit transport over MPLS	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	In this feature, communication between the remote IPv6 domains runs native IPv6 protocols over a dedicated link, where the underlying mechanisms are fully transparent to IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> IPv6 over a Circuit Transport over MPLS, page 2 Deploying IPv6 over a Circuit Transport over MPLS, page 5
IPv6 using tunnels over the customer edge routers	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	Using tunnels on the CE routers is the simplest way of deploying IPv6 over MPLS networks with no impact on the operation or infrastructure of MPLS. The following sections provide information about this feature: <ul style="list-style-type: none"> IPv6 Using Tunnels on the Customer Edge Routers, page 3

Table 8 **Feature Information for Implementing IPv6 over MPLS (continued)**

Feature Name	Releases	Feature Information
IPv6 switching: provider edge router over MPLS (6PE)	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T	<p>The Cisco implementation of IPv6 provider edge router over MPLS enables IPv6 sites to communicate with each other over an MPLS IPv4 core network using MPLS LSPs.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • IPv6 on the Provider Edge Routers (6PE), page 4 • Deploying IPv6 on the Provider Edge Routers (6PE), page 5 • Specifying the Source Address Interface on a 6PE Router, page 7 • Binding and Advertising the 6PE Label to Advertise Prefixes, page 8 • Configuration Examples for IPv6 over MPLS, page 13
6PE multipath	12.2(25)S 12.2(28)SB 12.2(33)SRA 12.4(6)T	<p>The 6PE multipath feature uses multiprotocol internal BGP (MP-iBGP) to distribute IPv6 routes over the MPLS IPv4 core network and to attach an MPLS label to each route.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • 6PE Multipath, page 5 • Configuring iBGP Multipath Load Sharing, page 9

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003–2009 Cisco Systems, Inc. All rights reserved.



Implementing IPv6 VPN over MPLS

First Published: February 28, 2007

Last Updated: November 11, 2008

The Border Gateway Protocol (BGP)-Multiprotocol Label Switching (MPLS) virtual private network (VPN) feature represents an implementation of the provider edge (PE)-based VPN model. This document describes the IPv6 VPN over MPLS (6VPE) feature.

In principle, there is no difference between IPv4 VPNs and IPv6 VPNs. In both IPv4 and IPv6, multiprotocol BGP is the centerpiece of the MPLS VPN for IPv6 (VPNv6) architecture. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing IPv6 VPN over MPLS” section on page 59](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing IPv6 VPN over MPLS, page 2](#)
- [Restrictions for Implementing IPv6 VPN over MPLS, page 2](#)
- [Information About Implementing IPv6 VPN over MPLS, page 2](#)
- [How to Implement IPv6 VPN over MPLS, page 9](#)
- [Configuration Examples for Implementing IPv6 VPN over MPLS, page 54](#)
- [Additional References, page 55](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

- [Command Reference, page 56](#)
- [Feature Information for Implementing IPv6 VPN over MPLS, page 59](#)
- [Glossary, page 60](#)

Prerequisites for Implementing IPv6 VPN over MPLS

Your network must be running the following Cisco IOS services before you configure IPv6 VPN operation:

- MPLS in provider backbone routers
- MPLS with VPN code in provider routers with VPN PE routers
- BGP in all routers providing a VPN service
- Cisco Express Forwarding switching in every MPLS-enabled router
- Class of Service (CoS) feature

Restrictions for Implementing IPv6 VPN over MPLS

6VPE supports an MPLS IPv4-signaled core. An MPLS IPv6-signaled core is not supported.

Information About Implementing IPv6 VPN over MPLS

Multiprotocol BGP is the centerpiece of the MPLS IPv6 VPN architecture in both IPv4 and IPv6. It is used to distribute IPv6 routes over the service provider backbone, using the same procedures to work with overlapping addresses, redistribution policies, and scalability issues.

Although IPv6 should not have overlapping address space, IPv6 addresses are prepended with a route distinguisher (RD). A network layer reachability information (NLRI) 3-tuple format (which contains length, IPv6 prefix, and label) is defined to distribute these routes using multiprotocol BGP. The extended community attribute—the route target—is used to control redistribution of routing information by tagging exported routes and filtering imported ones.

For scalability, route reflectors can be used to concentrate routing paths and avoid a full PE mesh. BGP features in IPv6, such as route refresh, automatic route filtering, and outbound route filtering, help reduce the number of routes held in each PE.

This document focuses on the following differences between IPv4 and IPv6:

- Creation of a new multiprotocol BGP IPv6 VPN address family and specification of a IPv6 VPN address format
- Specification of a new IPv6 VPN NLRI
- Specification of BGP next-hop encoding when the router has an IPv4-based MPLS core

Some IPv6 VPN features, such as interprovider and Carrier Supporting Carrier (CSC) topologies, are specific to BGP-MPLS IPv6 VPN. For instance, the link between Autonomous System Boundary Routers (ASBRs) might support IPv4 only, IPv6 only, or both, independently of the address family being transported.

Before you configure IPv6 VPN over MPLS, you should understand the following concepts:

- [Addressing Considerations for IPv6 VPN over MPLS \(6VPE\), page 3](#)
- [Basic IPv6 VPN over MPLS Functionality, page 3](#)
- [Advanced IPv6 MPLS VPN Functionality, page 6](#)

Addressing Considerations for IPv6 VPN over MPLS (6VPE)

Regardless of the VPN model deployed (such as customer edge [CE]-based, PE-based), an addressing plan must be defined for the VPN that allows hosts to communicate with other sites using one site within one VPN, and with public resources.

VPN IPv4 sites often use private addressing for their addressing plan. These addresses need not be registered, and they are not routable on the public network. Whenever a host within a private site needs to access a public domain, it goes through a device that finds a public address on its behalf. With IPv4, this can be a network address translator or an application proxy.

Given the larger address space available with IPv6, the easiest approach to IPv6 addressing is to use IPv6 global addresses for the private addressing plan. Another approach is to use unique local addresses (ULAs). ULAs are easy to filter at site boundaries based on their scope. ULAs are also Internet service provider (ISP)-independent and can be used for communications inside a site without any permanent or intermittent Internet connectivity.

In 6VPE, ULAs are treated as regular global addresses. The router configuration filters ULA prefixes to prevent them from appearing in the public domain. Link-local addresses on the peer will not be announced by BGP (IPv6 or IPv6 VPN) speakers.

A host within a private site that needs to access a public domain can do so through an IPv6 application proxy (such as a web proxy for accessing web pages), which accesses the public resource on the host's behalf with a global routable address, or the host can use a public address of its own. In the latter case, if ULAs have been deployed, the IPv6 host also is configured with a routable global address. A source address selection algorithm is used to select one or the other, based on the destination address.

Basic IPv6 VPN over MPLS Functionality

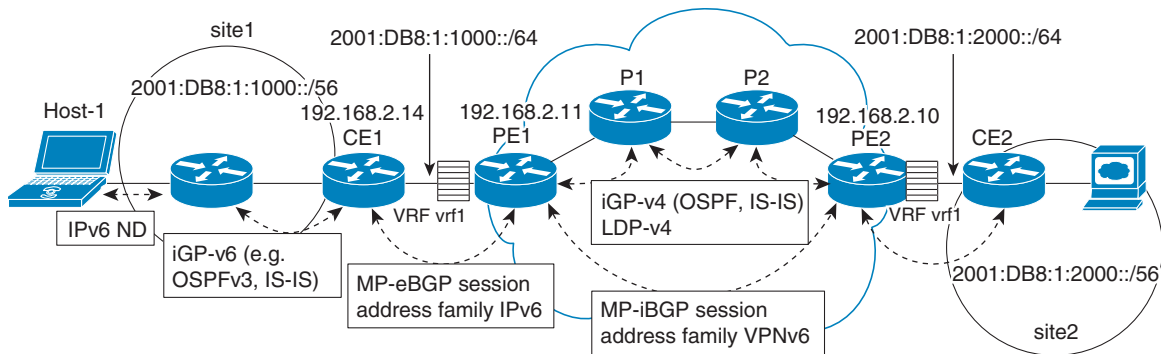
The transition to IPv6 is expected to lead to a long period of coexistence between IPv4 and IPv6. A method for deploying IPv6 VPN takes advantage of this coexistence by leveraging an existent MPLS IPv4 core network. This approach is called 6VPE.

The following sections describe concepts for basic IPv6 MPLS VPN functionality:

- [IPv6 VPN Architecture Overview, page 3](#)
- [IPv6 VPN Next Hop, page 4](#)
- [MPLS Forwarding, page 5](#)
- [VRF Concepts, page 5](#)
- [IPv6 VPN Scalability, page 6](#)

IPv6 VPN Architecture Overview

Figure 1 illustrates the important aspects of the IPv6 VPN architecture.

Figure 1 Simple IPv6 VPN Architecture

The CE routers are connected to the provider's backbone using PE routers. The PE routers are connected using provider (P1 and P2 in Figure 1) routers. The provider (P) routers are unaware of VPN routes, and, in the case of 6VPE, may support only IPv4. Only PE routers perform VPN-specific tasks. For 6VPE, the PE routers are dual-stack (IPv4 and IPv6) routers.

The routing component of the VPN operation is divided into core routing and edge routing. Core routing, which involves PE routers and P routers, typically is performed by an IPv4 Interior Gateway Protocol (IGP) such as Open Shortest Path First (OSPF) or Intermediate System-to-Intermediate System (IS-IS). In Figure 1, the IGP distributes only routes internal to the provider's autonomous system. The core routing enables connectivity among P and PE routers.

Edge routing takes place in two directions: routing between PE pairs and routing between a PE and a CE. Routing between PE pairs is achieved using multiprotocol internal BGP (iBGP) using the IPv6 VPN address family. This method distributes routes learned from CEs through PE-CE routing, using appropriate route export policies at the ingress PE router and appropriate route import policies at the egress PE router.

Routing between the CE and its PE is achieved using a routing protocol that is VRF aware. Only static routes and external BGP (eBGP) are VPN routing and forwarding (VRF) instance aware. In Figure 1, eBGP is used between the CE (CE1) and the PE (PE1). At the same time, the CE runs an IPv6 IGP (such as OSPFv3 or IS-IS for IPv6) within the VPN site (site1 in Figure 1). The CE redistributes IGP routes into multiprotocol-eBGP address family IPv6. At the PE, these routes are installed in the VRF named vrf1, and forwarded to the remote PEs (PE2 in Figure 1), according to export policies defined for this VRF.

IPv6 VPN Next Hop

When the router announces a prefix using the MP_REACH_NLRI attribute, MP-BGP running on one PE inserts a BGP next hop in the update message sent to a remote PE. This next hop is either propagated from the received update (for instance, if the PE is a route reflector), or it is the address of the PE sending the update message (the egress PE).

For the IPv6 VPN address family, the next hop must be an IPv6 VPN address, regardless of the nature of the network between the PE speakers. Because the RD has no significance (the address is not part of any VPN), it is set to 0. If the provider network is a native IPv6 network, the remaining part of the next hop is the IPv6 address of the egress PE. Otherwise, it is an IPv4 address used as an IPv6-mapped address (for example, ::FFFF:IPv4-address).

See the “[IPv6 VPN Configuration Using IPv4 Next Hop: Example](#)” section on page 54 for an example of IPv6 VPN next-hop configuration.

MPLS Forwarding

Upon receiving IPv6 traffic from one customer site, the ingress PE router uses MPLS to tunnel IPv6 VPN packets over the backbone toward the egress PE router identified as the BGP next hop. The ingress PE router typically prepends the IPv6 packets with the outer and inner labels before putting the packet on the egress interface.

Under normal operation, a P router along the forwarding path does not look inside the frame beyond the first label. The P router either swaps the incoming label with an outgoing one, or removes the incoming label if the next router is a PE router. Removing the incoming label is called penultimate hop popping. The remaining label (BGP label) is used to identify the egress PE interface toward the customer site. It also hides the protocol version (IPv6) from the last P router, which would otherwise need to forward an IPv6 packet.

A P router is ignorant about IPv6 VPN routes. The IPv6 header remains hidden under one or more MPLS labels. When the P router receives an MPLS-encapsulated IPv6 packet that cannot be delivered, it has two options. If the P router is IPv6 aware, it exposes the IPv6 header, builds an Internet Control Message Protocol (ICMP) for IPv6 message, and sends the message, which is MPLS encapsulated, to the source of the original packet. If the P router is not IPv6 aware, it drops the packet.

6VPE over GRE Tunnels

In some Cisco IOS releases, the ingress PE router uses IPv4 generic routing encapsulation (GRE) tunnels combined with 6VPE over MPLS to tunnel IPv6 VPN packets over the backbone toward the egress PE router identified as the BGP next hop.

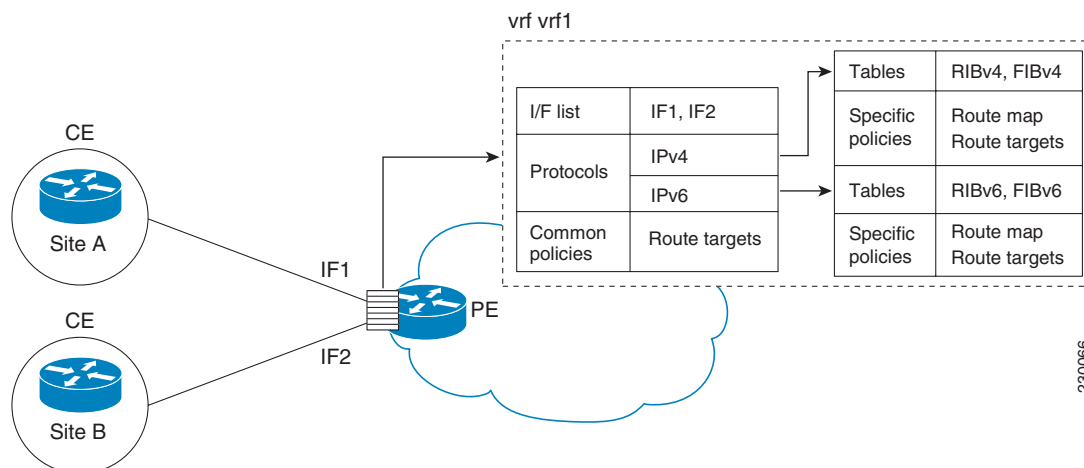
VRF Concepts

A VRF is a virtual routing and forwarding entity that works with a private customer-specific Routing Information Base (RIB) and Forwarding Information Base (FIB). Although IPv4 and IPv6 routing tables are distinct, it is convenient for the two protocols to share the same VRF for a specific customer.

IPv6 VPN customers are likely to be existing VPNv4 customers that are either deploying dual-stack hosts and routers or shadowing some of their IPv4 infrastructure with IPv6 nodes. Several deployment models are possible. Some customers use separate logical interfaces for IPv4 and IPv6 and define separate VRFs on each. Although this approach provides flexibility to configure separate policies for IPv4 and IPv6, it prevents sharing the same policy. Another approach, the multiprotocol VRF, keeps a single VRF on the PE-CE interface, and enables it for IPv4, IPv6, or both. It is then possible to define common or separate policies for each IP version. With this approach, a VRF is better defined as the set of tables, interfaces, and policies found at the PE, and is used by sites of a particular VPN connected to this PE.

[Figure 2](#) illustrates the multiprotocol VRF, in which the VRF named `vrf1` is enabled for both IPv4 and IPv6 and is associated with two interfaces (IF1, IF2), two sets of tables (IPv4 RIB and FIB and IPv6 RIB and FIB), and a set of common or distinct policies.

For information on how to configure a VRF in IPv6, see the [“Configuring a Virtual Routing and Forwarding Instance for IPv6”](#) section on [page 9](#).

Figure 2 **Multiprotocol VRF**

IPv6 VPN Scalability

PE-based VPNs such as BGP-MPLS IPv6 VPN scale better than CE-based VPNs. A network designer must consider scaling when designing the network. Scaling a BGP-MPLS IPv6 VPN is similar to scaling a BGP-MPLS IPv4 VPN. The following points need to be considered:

- Routing table size, which includes the size of VRF tables and BGP tables
- Number of BGP sessions, which grows as a square number of PEs

Routing table size concerns occur with PEs that handle many customer sites. Not only do these PEs have one RIB and FIB per connected customer, but also the PEs' BGP tables, which total all entries from individual VRFs, grow accordingly. Another scalability problem occurs when the number of PEs in the provider network grows beyond a certain level. Assuming that a significant number of sites belonging to the same VPN are spread over many PEs, the number of multiprotocol BGP sessions may rapidly become prohibitive: $(n-1) \times n/2$, where n is the number of PEs.

The following features are included in IPv6 VPN over MPLS:

- Route refresh and automatic route filtering—Limits the size of routing tables, because only routes imported into a VRF are kept locally. When the import policy changes, a route refresh can be sent to query a retransmission of routing updates.
- Outbound route filtering (ORF)—Allows the ingress PE to advertise filters to the egress PE so that updates are not sent unnecessarily over the network.
- Route reflectors—Route reflectors (RRs) are iBGP peers that propagate iBGP routes learned from other iBGP peers. RRs are used to concentrate iBGP sessions.

Advanced IPv6 MPLS VPN Functionality

Advanced MPLS features such as accessing the Internet from a VPN for IPv4, multiautonomous-system backbones, and CSCs are generally the same for IPv6 as for IPv4. However, there are differences in addressing and in the way 6VPE operates over an IPv4 backbone.

The following sections describe concepts for advanced IPv6 MPLS VPN functionality:

- [Internet Access, page 7](#)

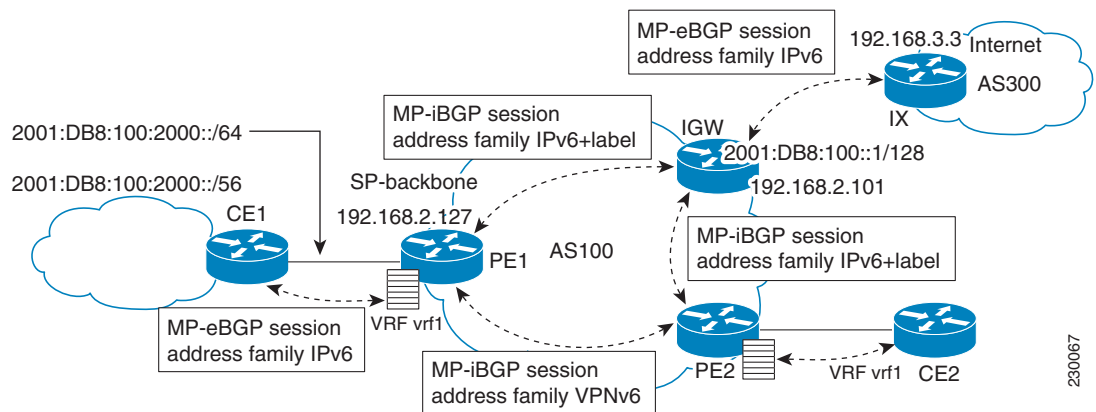
- [Multiautonomous-System Backbones](#), page 7
- [Carrier Supporting Carriers](#), page 8

Internet Access

Most VPN sites require access to the Internet. RFC 4364 describes a set of models for enabling VPN access to the Internet. All these models apply to IPv6 VPNs as well. In one approach, one interface is used by the CE to connect to the Internet and a different one to connect to the VRF. Another model is in which all Internet routes are redistributed into the VRF. This approach has the disadvantage of requiring the Internet routes to be replicated in each VRF.

In one scenario, a static route is inserted into the VRF table, with a next hop that points to the Internet gateway found in the IPv6 default table. [Figure 3](#) illustrates this scenario, in which Internet access is provided to the customer in the VRF named `vrf1`.

Figure 3 **Internet Access Topology**



For a customer site (site1 in [Figure 3](#)) to access public resources over the Internet, this site must be known by a public prefix. Unlike IPv4, IPv6 does not offer a Network Address Translation (NAT) mechanism that allows translating private addresses into public addresses when leaving the site boundaries. Not only does that imply that hosts within the site speak with public addresses, but also that these addresses (or the prefix they belong to) must appear in the public domain.

For outbound traffic, the default route configured in the VRF table at ingress PE (PE1) directs traffic for destinations outside the VPN to the Internet gateway.

For inbound traffic, a route must exist at the Internet gateway to direct the traffic for customer site via its PE of attachment (PE1 in [Figure 3](#)). This route can be distributed by the ingress PE (PE1) using multiprotocol iBGP (with the IPv6 address family configuration), so no specific configuration needs to be done on a per VPN PE basis at the Internet gateway. Nevertheless, for inbound traffic at PE1, a route must exist in the default table for the customer site global prefix pointing to the VRF of the site.

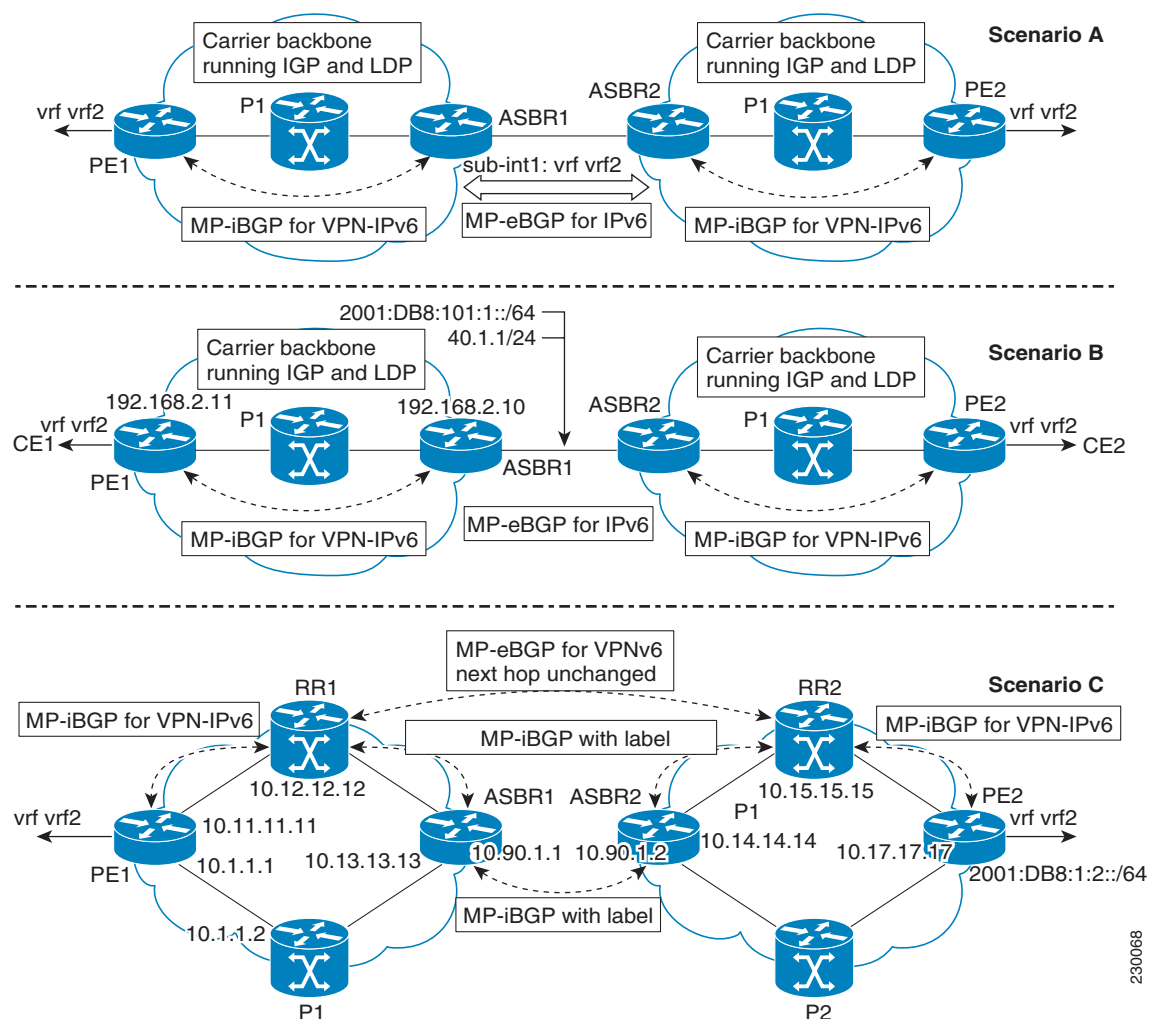
Multiautonomous-System Backbones

The problem of interprovider VPNs is similar for IPv6 and IPv4, assuming that IPv6 was deployed everywhere IPv4 was deployed.

In IPv6 deployments that cross autonomous system boundaries, providers may have to obtain a peering model, or work with the peering model put in place for VPNv4.

Figure 4 illustrates interprovider scenarios in IPv6 VPN.

Figure 4 *Interprovider Scenarios*



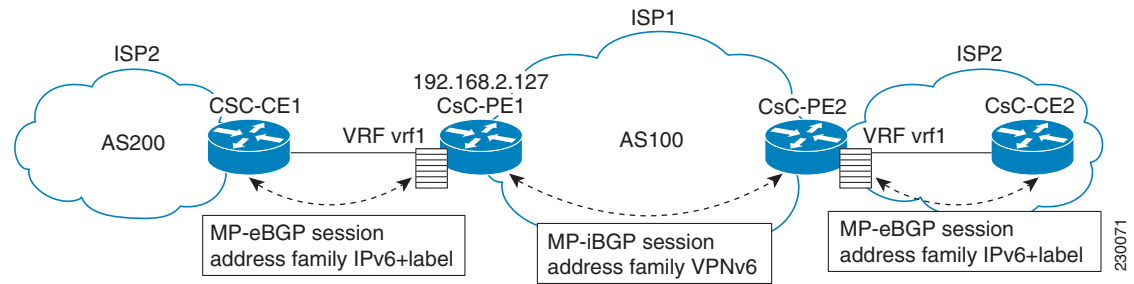
Depending on the network protocol used between ASBRs, the three scenarios shown in Figure 4 can have several implementation options. For instance, scenario B, which suggests a multiprotocol eBGP IPv6 VPN peering between ASBRs, could use either an IPv6 or an IPv4 link.

In scenario C, multihop multiprotocol eBGP redistributes IPv6 VPN routes across route reflectors in different autonomous systems. Labeled IPv4 routes to the PEs (in the 6VPE case) need to be advertised across ASBRs so that a complete labeled switch path is set up end to end.

Carrier Supporting Carriers

The CSC feature provides VPN access to a customer service provider, so this service needs to exchange routes and send traffic over the ISP MPLS backbone. The only difference from a regular PE is that it provides MPLS-to-MPLS forwarding on the CSC-CE to CSC-PE interface, rather than IP-to-MPLS forwarding.

Figure 5 highlights the two ISPs' interface.

Figure 5 *CSC 6VPE Configuration Example*

For information on configuring CSC for BGP-MPLS VPN for IPv6, see the [“Configuring CSC for IPv6 VPN”](#) section on page 47.

How to Implement IPv6 VPN over MPLS

The following sections provide information on how to implement IPv6 VPN over MPLS:

- [Configuring a Virtual Routing and Forwarding Instance for IPv6, page 9](#)
- [Binding a VRF to an Interface, page 11](#)
- [Configuring a Static Route for PE-to CE-Routing, page 12](#)
- [Configuring eBGP PE-to-CE Routing Sessions, page 13](#)
- [Configuring the IPv6 VPN Address Family for iBGP, page 14](#)
- [Configuring Route Reflectors for Improved Scalability, page 16](#)
- [Configuring Internet Access, page 22](#)
- [Configuring a Multiautonomous-System Backbone for IPv6 VPN, page 30](#)
- [Configuring CSC for IPv6 VPN, page 47](#)
- [Verifying and Troubleshooting IPv6 VPN, page 48](#)

Configuring a Virtual Routing and Forwarding Instance for IPv6

A VRF is an address family-independent object that can be enabled and configured for each of the supported address families. Configuring a VRF consists of the following three steps:

1. Configuring the address-family-independent part of the VRF
2. Enabling and configuring IPv4 for the VRF
3. Enabling and configuring IPv6 for the VRF

A VRF is given a name and an RD. The RD is configured outside the context of the address family, although the RD is used to distinguish overlapping addresses within the context of a particular BGP address family. Having separate RDs for IPv4 VPN addresses and IPv6 VPN addresses does not matter. On Cisco routers, the RDs are the same in order to simplify configuration and VPN management.

Users can configure policies in common between IPv4 and IPv6 when not using an address family context. This feature is shared route targets (import and export), and it is useful in a migration scenario, where IPv4 policies already are configured and IPv6 policies should be the same as the IPv4 policies.

The IPv4 and IPv6 address family can each be enabled and configured separately. Note that the route-target policies entered at this level override global policies that may have been specified during address family-independent configuration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mls ipv6 vrf**
4. **vrf definition** *vrf-name*
5. **rd** *route-distinguisher*
6. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
8. **route-target** {**import** | **export** | **both**} *route-target-ext-community*
9. **exit**
10. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
11. **route-target** {**import** | **export** | **both**} *route-target-ext-community*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	mls ipv6 vrf Example: Router(config)# mls ipv6 vrf	Enables IPv6 globally in a VRF.
Step 4	vrf definition <i>vrf-name</i> Example: Router(config)# vrf definition vrf1	Configures a VPN VRF routing table and enters VRF configuration mode.
Step 5	rd <i>route-distinguisher</i> Example: Router(config-vrf)# rd 100:1	Specifies the RD for a VRF.

	Command or Action	Purpose
Step 6	route-target { import export both } <i>route-target-ext-community</i> Example: Router(config-vrf)# route target import 100:10	Specifies the route target VPN extended communities for both IPv4 and IPv6.
Step 7	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Router(config)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 8	route-target { import export both } <i>route-target-ext-community</i> Example: Router(config-vrf-af)# route target import 100:11	Specifies the route target VPN extended communities specific to IPv4.
Step 9	exit Example: Router(config-vrf-af)# exit	Exits address family configuration mode on this VRF.
Step 10	address-family ipv6 [vrf vrf-name] [unicast multicast] Example: Router(config-vrf)# address-family ipv6	Enters address family configuration mode for configuring routing sessions such as BGP that use standard IPv6 address prefixes.
Step 11	route-target { import export both } <i>route-target-ext-community</i> Example: Router(config-vrf-af)# route target import 100:12	Specifies the route target VPN extended communities specific to IPv6.

Binding a VRF to an Interface

The following task describes how to bind a VRF to an interface. In order to specify which interface belongs to which VRF, use the **vrf forwarding** command for both IPv4 and IPv6. An interface cannot belong to more than one VRF. When the interface is bound to a VRF, previously configured addresses (IPv4 and IPv6) are removed, and they must be reconfigured.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **vrf forwarding** *vrf-name*
5. **ip address** *ip-address mask* [**secondary**]

6. `ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 4	vrf forwarding <i>vrf-name</i> Example: Router(config-if)# vrf forwarding vrf1	Associates a VPN VRF with an interface or subinterface. Note that any address, IPv4 or IPv6, that was configured prior to entering this command will be removed.
Step 5	ip address <i>ip-address mask [secondary]</i> Example: Router(config-if)# ip address 10.10.10.1 255.255.255.0	Configures an IPv4 address on the interface.
Step 6	ipv6 address <i>{ipv6-address/prefix-length prefix-name sub-bits/prefix-length}</i> Example: Router(config-if)# ipv6 address 2001:DB8:100:1::1/64	Configures an IPv6 address on the interface.

Configuring a Static Route for PE-to CE-Routing

The following task describes how to configure a static route for PE-to-CE routing.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 route** [*vrf vrf-name*] *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [**nexthop-vrf** [*vrf-name1* | **default**]] [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [*next-hop-address*] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route [vrf <i>vrf-name</i>] <i>ipv6-prefix/prefix-length</i> [<i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>]] [nexthop-vrf [<i>vrf-name1</i> default]] [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>next-hop-address</i>] [tag <i>tag</i>] Example: Router(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:200::1 nexthop-vrf default	Installs the specified IPv6 static route using the specified next hop.

Configuring eBGP PE-to-CE Routing Sessions

The following task describes how to configure eBGP PE-to-CE routing sessions.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast] Example: Router(config-router)# address-family ipv6 vrf vrf1	Enters address family configuration mode.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router-af)# neighbor 2001:DB8:100:1::2 remote-as 200	Adds an entry to the multiprotocol BGP neighbor table.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 2001:DB8:100:1::2 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

Configuring the IPv6 VPN Address Family for iBGP

The following task describes how to configure the IPv6 VPN address family for iBGP.

SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *autonomous-system-number*
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*

6. **address-family vpnv6 [unicast]**
7. **neighbor {ip-address | peer-group-name | ipv6-address} activate**
8. **neighbor {ip-address | ipv6-address | peer-group-name} send-community [both | standard | extended]**
9. **exit**

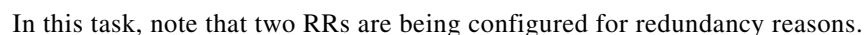
DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp autonomous-system-number Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 192.168.2.11 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table. <ul style="list-style-type: none"> In IPv6 VPN, the peer address typically is an IPv4 address, in order to enable the BGP session to be transported over the IPv4-based core network.
Step 5	neighbor {ip-address ipv6-address peer-group-name} update-source interface-type interface-number Example: Router(config-router)# neighbor 192.168.2.11 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Router(config-router)# address-family vpnv6	Places the router in address family configuration mode for configuring routing sessions.
Step 7	neighbor {ip-address peer-group-name ipv6-address} activate Example: Router(config-router-af)# neighbor 192.168.2.11 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

Configuring Route Reflectors for Improved Scalability

In an MPLS-based core, RRs are not part of the label switch paths and can be located anywhere in the network. For example, in a flat RR design, RRs can be deployed at Level 1 points of presence (POPs) and peer together in a full-mesh topology. In a hierarchical RR design, RRs could be deployed at Level 1 and Level 2 POPs, with Level 1 POPs peering together and with Level 2 RRs.

Figure 6 **Route Reflector Peering Design**



The following list of BGP RR clients must be configured at each IPv6 RR (RR6 and RR6_1 in [Figure 6](#)) router, at each POP:

- PE routers (PE-VPN) of the POP providing IPv6 VPN access to the ISP customers. This includes both IPv6 VPN (6VPE) peering for interconnecting customer sites and IPv6 peering (6PE) for providing Internet access to VPN customers (see the [“Configuring Internet Access”](#) section on [page 22](#)).
- Internet gateway (IGW) located in the POP in order to provide PE customers with access to the IPv6 Internet (see the [“Configuring Internet Access”](#) section on [page 22](#)).
- RRs from other service providers. This feature is used to provide interautonomous-system connectivity, and it includes both IPv6 and IPv6 VPN peering. This service is described in the [“Configuring a Multiautonomous-System Backbone for IPv6 VPN”](#) section on [page 30](#) section.
- RRs in other POPs. All RRs peer together, with both IPv6 and IPv6 VPN address families enabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
11. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*tll*]
13. **address-family ipv6**
14. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
15. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
16. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
17. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
18. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
19. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
20. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
21. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
22. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
23. **exit**

24. **address-family vpnv6 [unicast]**
25. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
26. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
27. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
28. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
29. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
30. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
31. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
32. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
33. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
34. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.101 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the Internet gateway in order to provide Internet access.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.101 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.

	Command or Action	Purpose
Step 6	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 192.168.2.121 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the other POP's RR.
Step 7	neighbor {ip-address ipv6-address peer-group-name} update-source interface-type interface-number Example: Router(config-router)# neighbor 192.168.2.121 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 8	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 192.168.2.127 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table.
Step 9	neighbor {ip-address ipv6-address peer-group-name} update-source interface-type interface-number Example: Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 10	neighbor {ip-address ipv6-address peer-group-name} remote-as as-number Example: Router(config-router)# neighbor 192.168.2.1 remote-as 200	(Optional) Adds an entry to the multiprotocol BGP neighbor table, and provides peering with the RR of the peer ISP in order to provide inter-VPN service.
Step 11	neighbor {ip-address ipv6-address peer-group-name} update-source interface-type interface-number Example: Router(config-router)# neighbor 192.168.2.1 update-source Loopback 0	(Optional) Enables the BGP session to use a source address on the specified interface.
Step 12	neighbor {ip-address ipv6-address peer-group-name} ebgp-multihop [ttl] Example: Router(config-router)# neighbor 192.168.2.1 ebgp-multihop	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 13	address-family ipv6 Example: Router(config-router)# address-family ipv6	(Optional) Enters address family configuration mode in order to provide Internet access service.

	Command or Action	Purpose
Step 14	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.101 activate	(Optional) Enables the exchange of information for this address family with the specified neighbor.
Step 15	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.101 send-label	(Optional) Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.
Step 16	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.101 route-reflector-client	(Optional) Configures the router as a BGP route reflector and configures the specified neighbor as its client.
Step 17	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.121 activate	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.
Step 18	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.121 send-label	(Optional) Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.
Step 19	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.121 route-reflector-client	(Optional) Configures the specified neighbor as a route reflector client.
Step 20	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.127 activate	(Optional) Enables the exchange of information for this address family with the specified BGP neighbor.
Step 21	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.127 send-label	(Optional) Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router.

	Command or Action	Purpose
Step 22	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client	(Optional) Configures the specified neighbor as a route reflector client.
Step 23	exit Example: Router(config-router-af)# exit	(Optional) Exits address family configuration mode.
Step 24	address-family vpnv6 [unicast] Example: Router(config-router)# address-family vpnv6	Places the router in address family configuration mode for configuring routing sessions.
Step 25	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.121 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 26	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.21 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 27	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.121 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 28	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 29	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.127 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.

	Command or Action	Purpose
Step 30	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 31	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.1 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 32	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.1 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 33	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.1 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 34	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } next-hop-unchanged [allpaths] Example: Router(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths	Enables an EBGp multihop peer to propagate to the next hop unchanged for paths.

Configuring Internet Access

Most VPN customers have access to the IPv4 Internet, and customers that get IPv6 VPN access will need to have access to the IPv6 Internet. The design of this service is similar to a global Internet access service. 6VPE routers located in a Level 1 POP (co-located with an IGW router) can access the IGW natively, whereas 6VPE routers located in Level 2 and Level 3 POPs with no direct access to the IGW can access the IGW in their closest Level 1 POP over 6PE.

Configuring VPN Internet access in such a 6VPE router involves configuring BGP peering with the IGW (in most cases through the IPv6 RR, as described in the [“Configuring Route Reflectors for Improved Scalability”](#) section). Then the user must configure cross-table routing to enable communication between the private domain (the VRF) and the public domain (the Internet).

Figure 3 is used to illustrate the following configuration tasks:

- [Configuring the Internet Gateway, page 23](#)
- [Configuring the IPv6 VPN PE, page 26](#)

Configuring the Internet Gateway

Configuring the Internet gateway for Internet access consists of the following tasks:

- [Configuring iBGP 6PE Peering to the VPN PE, page 23](#)
- [Configuring the Internet Gateway as the Gateway to the Public Domain, page 24](#)
- [Configuring eBGP Peering to the Internet, page 25](#)

Configuring iBGP 6PE Peering to the VPN PE

This task describes how to configure iBGP 6PE peering to the VPN PE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv6**
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	router bgp <i>autonomous-system-number</i>	Configures the BGP routing process.
	Example: Router(config)# router bgp 100	
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i>	Adds an entry to the multiprotocol BGP neighbor table to provide peering with the VPN PE.
	Example: Router(config-router)# neighbor 192.168.2.127 remote-as 100	

	Command or Action	Purpose
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type</i> <i>interface-number</i> Example: Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv6 Example: Router(config-router)# address-family ipv6	Enters address family configuration mode in order to exchange global table reachability.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.127 send-label	Enables a BGP router to send MPLS labels with BGP routes to a neighboring BGP router, and allows the PE VPN to reach the Internet gateway over MPLS.

Configuring the Internet Gateway as the Gateway to the Public Domain

Using the 6PE peering configuration established in the [“Configuring iBGP 6PE Peering to the VPN PE” section on page 23](#), this task describes how to configure the gateway to be the gateway to the public domain.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family** **ipv6**
5. **network** *ipv6-address/prefix-length*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	address-family ipv6 Example: Router(config-router)# address-family ipv6	Enters address family configuration mode in order to exchange global table reachability.
Step 5	network <i>ipv6-address/prefix-length</i> Example: Router(config-router-af)# network 2001:DB8:100::1/128	Configures the network source of the next hop to be used by the PE VPN.
Step 6	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring eBGP Peering to the Internet

The following task describes how to configure eBGP peering to the Internet to populate the global table.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **address-family ipv6**
6. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
7. **aggregate-address** *address mask* [**as-set**] [**summary-only**] [**suppress-map** *map-name*] [**advertise-map** *map-name*] [**attribute-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor FE80::300::1%Ethernet0/0 remote-as 300	Adds an entry to the multiprotocol BGP neighbor table, and provides peering with PE (PE-VPN). <ul style="list-style-type: none"> Note that the peering is done over link-local addresses.
Step 5	address-family ipv6 Example: Router(config-router)# address-family ipv6	Enters address family configuration mode in order to exchange global table reachability.
Step 6	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor FE80::300::1%Ethernet0/0 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 7	aggregate-address <i>address mask</i> [as-set] [summary-only] [suppress-map <i>map-name</i>] [advertise-map <i>map-name</i>] [attribute-map <i>map-name</i>] Example: Router(config-router-af)# aggregate-address 2001:DB8::/32 summary-only	Creates an aggregate prefix before advertising it to the Internet.

Configuring the IPv6 VPN PE

Configuring the IPv6 VPN PE for Internet access consists of the following tasks:

- Configuring a Default Static Route from the VRF to the Internet Gateway, page 27
- Configuring a Static Route from the Default Table to the VRF, page 27
- Configuring iBGP 6PE Peering to the Internet Gateway, page 28

Configuring a Default Static Route from the VRF to the Internet Gateway

The following task describes how to configure a default static route from the VRF to the IGW.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *[vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 | default]] [administrative-distance] [administrative-multicast-distance | unicast | multicast] [next-hop-address] [tag tag]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route <i>[vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 default]] [administrative-distance] [administrative-multicast-distance unicast multicast] [next-hop-address] [tag tag]</i> Example: Router(config)# ipv6 route vrf vrf1 ::/0 2001:DB8:100::1 nexthop-vrf default	Configures a default static route from the VRF to the Internet gateway in order to allow outbound traffic to leave the VRF.

Configuring a Static Route from the Default Table to the VRF

The following task describes how to configure a static route from the default table to the VRF.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *[vrf vrf-name] ipv6-prefix/prefix-length {ipv6-address | interface-type interface-number [ipv6-address]} [nexthop-vrf [vrf-name1 | default]] [administrative-distance] [administrative-multicast-distance | unicast | multicast] [next-hop-address] [tag tag]*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route [vrf <i>vrf-name</i>] <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>]} [nexthop-vrf [<i>vrf-name1</i> default]] [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>next-hop-address</i>] [tag <i>tag</i>] Example: Router(config)# ipv6 route 2001:DB8:100:2000::/64 nexthop-vrf vrf1	Configures a static route from the default table to the VRF in order to allow inbound traffic to reach the VRF.

Configuring iBGP 6PE Peering to the Internet Gateway

The following task describes how to configure iBGP 6PE peering to the Internet gateway.

SUMMARY STEPS

- enable**
- configure terminal**
- router bgp** *autonomous-system-number*
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
- address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
- neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
- neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
- network** *ipv6-address/prefix-length*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.101 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the Internet gateway.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.101 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv6 [<i>vrf vrf-name</i>] [unicast multicast] Example: Router(config-router)# address-family ipv6	Enters address family configuration mode in order to exchange global table reachability.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.101 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor {ip-address ipv6-address peer-group-name} send-label Example: Router(config-router-af)# neighbor 192.168.2.101 send-label	Enables label exchange for this address family to this neighbor in order to enable the VPN PE to reach the Internet gateway over MPLS.
Step 9	network ipv6-address/prefix-length Example: Router(config-router-af)# network 2001:DB8:100:2000::/64	Provides the VRF prefix to the Internet gateway.

Configuring a Multiautonomous-System Backbone for IPv6 VPN

Two sites of a VPN may be connected to different autonomous systems because, for instance, the sites are connected to different service providers. The PE routers attached to that VPN would then be unable to maintain iBGP connections with each other or with a common route reflector. In a situation such as this one, there needs to be some way to use eBGP to distribute VPN-IPv6 addresses.

The following configuration example illustrates two scenarios, one in which a multiprotocol eBGP-IPv6 VPN peering between ASBRs uses an IPv4 link, and the same scenario using an IPv6 link. If the peering between ASBRs is performed over an IPv4 link, the BGP configuration on ASBR1 is as follows:

```
router bgp 1001
 no bgp default ipv4-unicast
 no bgp default route-target filter
 neighbor 192.1.1.1 remote-as 1002
 neighbor 192.168.2.11 remote-as 1001
 neighbor 192.168.2.11 update-source Loopback1
 !
 address-family vpnv6
 !Peering to ASBR2 over an IPv4 link
 neighbor 192.1.1.1 activate
 neighbor 192.1.1.1 send-community extended
 !Peering to PE1 over an IPv4 link
 neighbor 192.168.2.11 activate
 neighbor 192.168.2.11 next-hop-self
 neighbor 192.168.2.11 send-community extended
```

If the peering between ASBRs is performed over an IPv6 link, the BGP configuration on ASBR1 is as follows:

```
router bgp 1001
 neighbor 2001:DB8:101::72d remote-as 1002
 !
 address-family vpnv6
 !Peering to ASBR2 over an IPv6 link
 neighbor 2001:DB8:101::72d activate
 neighbor 2001:DB8:101::72d send-community extended
```

The next several tasks describe how to configure the PE VPN for a multiautonomous-system backbone using multihop multiprotocol eBGP to redistribute VPN routes across RRs in different autonomous systems. Labeled IPv4 routes to the PEs are advertised across ASBRs so that a complete label switch path (LSP) is set up end to end.

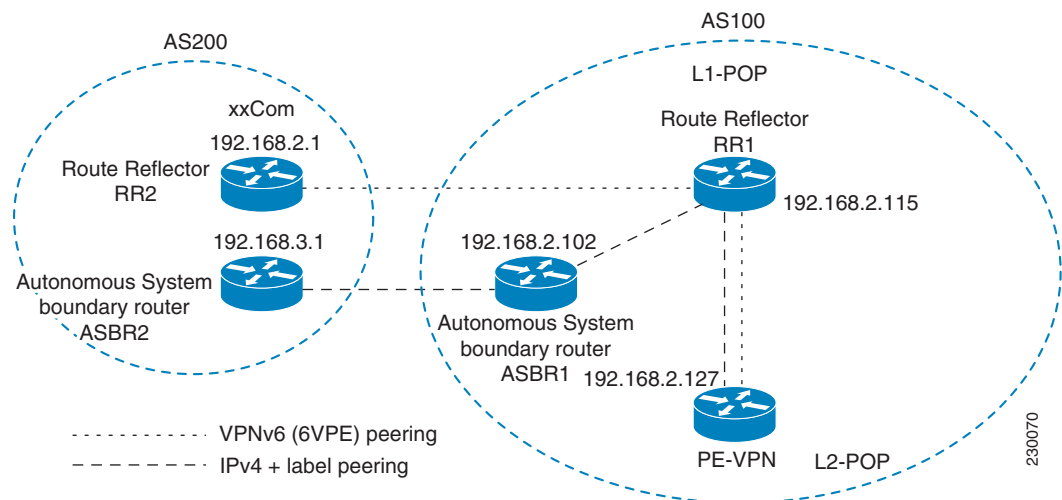
In this scenario, the ASBRs are not VPN aware; only the RRs are VPN aware. The following configuration should be available and understood:

- The ASBRs are providing the PE's loopback addresses to service providers they peer with. That includes:
 - The VPN PE's IPv4 loopback address (/32) for enabling next-hop resolution at the remote service provider location.
 - The VPN RR's IPv4 loopback address (/32) for enabling interprovider (interRR) eBGP peering.
- For VPN PE's IPv4 loopback address, the address providing is performed over multiprotocol BGP, with label, up to the remote PEs, so that the label establishes an end-to-end LSP. Therefore, the following MP-BGP peering was set up for VPNv4:
 - VPN PEs are iBGP peering with VPN RRs
 - ASBRs are iBGP peering with VPN RRs
 - ASBRs are eBGP peering with the remote service provider ASBR
- The VPN RRs of each service provider are peering together over eBGP and exchanging VPN routes. The next hop is forwarded unchanged, so that the end-to-end LSP is not via RRs.

To enable IPv6 VPN interautonomous-system access in this scenario, the ISP needs to modify the configurations at the PE VPN and at the RR. The same RRs are set up to provide a similar service for VPNv4. In that context, because the peering between the RR and the ASBR and between ASBRs is solely to exchange labels for IPv4 next hops used by both IPv4 VPN and IPv6 VPN, the ASBRs remain completely IPv6 unaware, and no configuration change is required there.

Figure 7 shows the BGP peering points required to enable IPv6 interprovider connectivity from the PE-VPN router (providing IPv6 VPN access) to the xxCom network.

Figure 7 BGP Peering Points for Enabling InterAS Scenario C



The following list of additional BGP peerings is necessary to enable interautonomous-system communication from the IPv6 VPN PE located in the Level 2 POP:

- IPv4 with label peering from the PE VPN to the route reflector named RR1 (which is already configured if VPNv4 interAS is deployed on the same nodes, using the same LSP).
- IPv4 with label peering from RR1 to ASBR1.

- IPv4 with label peering between ASBR1 and ASBR1.
- IPv6 VPN peering between RR1 and RR2 (which is the route reflector in the other autonomous systems) to exchange IPv6 VPN routes.
- IPv6 VPN peering with RR1. If the same route reflectors used to scale the IPv6 VPN service are used for interautonomous-system capability, then this function might also be already configured (see the “[Configuring Route Reflectors for Improved Scalability](#)” section on page 16).

Configuring the multiautonomous-system backbone for IPv6 VPN consists of the following steps:

1. [Configuring the PE VPN for a Multiautonomous-System Backbone, page 32](#)
2. [Configuring the Route Reflector for a Multiautonomous-System Backbone, page 35](#)
3. [Configuring the ASBR, page 43](#)

Configuring the PE VPN for a Multiautonomous-System Backbone

Configuring the PE VPN for a multiautonomous-system backbone consists of the following tasks:

- [Configuring iBGP IPv6 VPN Peering to a Route Reflector, page 32](#)
- [Configuring IPv4 and Label iBGP Peering to a Route Reflector, page 34](#)

Configuring iBGP IPv6 VPN Peering to a Route Reflector

The following task describes how to configure iBGP IPv6 VPN peering to a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector with interautonomous-system functionality.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Router(config-router)# address-family vpnv6	(Optional) Places the router in address family configuration mode for configuring routing sessions.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.115 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring IPv4 and Label iBGP Peering to a Route Reflector

The following task describes how to configure IPv4 and label iBGP peering to a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
5. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.

	Command or Action	Purpose
Step 4	address-family ipv4 [mdt multicast tunnel unicast [vrf vrf-name] vrf vrf-name] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 5	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to receive remote PE peer IPv4 loopback with label via RR1 in order to set up an end-to-end LSP.

Configuring the Route Reflector for a Multiautonomous-System Backbone

Configuring the route reflector for a multiautonomous-system backbone consists of the following tasks:

- [Configuring Peering to the PE VPN, page 35](#)
- [Configuring the Route Reflector, page 37](#)
- [Configuring Peering to the Autonomous System Boundary Router, page 40](#)
- [Configuring Peering to Another ISP Route Reflector, page 41](#)

Configuring Peering to the PE VPN

The following task describes how to configure peering to the provider edge VPN.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **exit**
10. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf vrf-name**] | **vrf vrf-name**]

11. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
12. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
13. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the route reflector for InterAS.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family <i>vpnv6</i> [unicast] Example: Router(config-router)# address-family vpnv6	(Optional) Places the router in address family configuration mode.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.115 send-community extended	Specifies that a community attribute should be sent to the BGP neighbor.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.
Step 10	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 11	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 12	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end LSP.
Step 13	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring the Route Reflector

The following task describes how to configure the route reflector.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family vpnv6** [**unicast**]

7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **route-reflector-client**
10. **exit**
11. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf vrf-name**] | **vrf vrf-name**]
12. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
13. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.127 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the VPN PE for InterAS.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.127 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family vpnv6 [unicast] Example: Router(config-router)# address-family vpnv6	(Optional) Places the router in address family configuration mode.

	Command or Action	Purpose
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [both standard extended] Example: Router(config-router-af)# neighbor 192.168.2.127 send-community extended	Specifies that a community attribute should be sent to the BGP neighbor.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } route-reflector-client Example: Router(config-router-af)# neighbor 192.168.2.127 route-reflector-client	Configures the specified neighbor as a route reflector client.
Step 10	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.
Step 11	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 12	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.127 activate	Enables the exchange of information for this address family with the specified neighbor.
Step 13	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.127 send-label	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end LSP.
Step 14	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring Peering to the Autonomous System Boundary Router

The following task describes how to configure peering to the Autonomous System Boundary Router (ASBR) named ASBR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [**vrf** *vrf-name*] | **vrf** *vrf-name*]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.102 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR1.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.102 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.

	Command or Action	Purpose
Step 6	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.102 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.102 send-label	Enables label exchange for this address family to this neighbor in order to receive the remote PE IPv4 loopback with the label set to an end-to-end LSP.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring Peering to Another ISP Route Reflector

The following task describes how to configure peering to another ISP route reflector, named RR2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family vpnv6** [**unicast**]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-community** [**both** | **standard** | **extended**]
10. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **next-hop-unchanged** [**allpaths**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.1 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for eBGP peering with RR2.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.1 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } ebgp-multihop [<i>ttl</i>] Example: Router(config-router)# neighbor 192.168.2.1 ebgp-multihop	(Optional) Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 7	address-family vpnv6 [unicast] Example: Router(config-router)# address-family vpnv6	(Optional) Places the router in address family configuration mode for configuring routing sessions.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.1 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-community [<i>both</i> <i>standard</i> <i>extended</i>] Example: Router(config-router-af)# neighbor 192.168.2.1 send-community extended	Specifies that a communities attribute should be sent to the BGP neighbor.
Step 10	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } next-hop-unchanged [<i>allpaths</i>] Example: Router(config-router-af)# neighbor 192.168.2.1 next-hop-unchanged allpaths	Enables an eBGP multihop peer to propagate to the next hop unchanged for paths.

Configuring the ASBR

Configuring the ASBR for a multiAS backbone consists of the following tasks:

- [Configuring Peering with Router Reflector RR1, page 43](#)
- [Configuring Peering with the Other ISP ASBR2, page 45](#)

Configuring Peering with Router Reflector RR1

The following task describes how to configure peering with a route reflector named RR1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **address-family ipv4** [*mdt* | *multicast* | *tunnel* | *unicast* [*vrf vrf-name*] | *vrf vrf-name*]
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.2.115 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with RR1.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.2.115 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	address-family ipv4 [mdt multicast tunnel unicast [vrf <i>vrf-name</i>] vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.2.115 activate	Enables the exchange of information for this address family with the specified BGP neighbor.

	Command or Action	Purpose
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.2.115 send-label	Enables label exchange for this address family to this neighbor in order to send to the local PE the remote PE IPv4 loopback with a label in order to set up an end-to-end LSP.
Step 9	exit Example: Router(config-router-af)# exit	Exits address family configuration mode.

Configuring Peering with the Other ISP ASBR2

The following task describes how to configure peering with the other ISP ASBR, ASBR2.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **router bgp** *autonomous-system-number*
4. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
5. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **update-source** *interface-type interface-number*
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **ebgp-multihop** [*ttl*]
7. **address-family ipv4** [**mdt** | **multicast** | **tunnel** | **unicast** [*vrf vrf-name*] | **vrf** *vrf-name*]
8. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
9. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**
10. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]
11. **network** {*network-number* [**mask** *network-mask*] | *nsap-prefix*} [**route-map** *map-tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 4	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router)# neighbor 192.168.3.1 remote-as 100	Adds an entry to the multiprotocol BGP neighbor table for peering with the ASBR2.
Step 5	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } update-source <i>interface-type interface-number</i> Example: Router(config-router)# neighbor 192.168.3.1 update-source Loopback 0	Enables the BGP session to use a source address on the specified interface.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } ebgp-multihop [<i>ttl</i>] Example: Router(config-router)# neighbor 192.168.3.1 ebgp-multihop	Accepts and attempts BGP connections to external peers residing on networks that are not directly connected.
Step 7	address-family ipv4 [mdt multicast tunnel unicast [<i>vrf vrf-name</i>] vrf <i>vrf-name</i>] Example: Router(config-router)# address-family ipv4	Enters address family configuration mode to configure a routing session using standard IPv4 address prefixes.
Step 8	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor 192.168.3.1 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 9	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor 192.168.3.1 send-label	Enables label exchange for this address family to this neighbor in order to receive the remote PE IPv4 loopback with a label in order to set up an end-to-end LSP.

	Command or Action	Purpose
Step 10	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Router(config-router-af)# network 192.168.2.27 mask 255.255.255.255	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the PE VPN loopback.
Step 11	network { <i>network-number</i> [mask <i>network-mask</i>] <i>nsap-prefix</i> } [route-map <i>map-tag</i>] Example: Router(config-router-af)# network 192.168.2.15 mask 255.255.255.255	Flags a network as local to this autonomous system and enters the network to the BGP table. This configuration is for the RR1 loopback.

Configuring CSC for IPv6 VPN

The following task shows how to configure CsC-PE1 peering configuration with CsC-CE1.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **hostname** *name*
4. **router bgp** *autonomous-system-number*
5. **address-family ipv6** [**vrf** *vrf-name*] [**unicast** | **multicast**]
6. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **remote-as** *as-number*
7. **neighbor** {*ip-address* | *peer-group-name* | *ipv6-address*} **activate**
8. **neighbor** {*ip-address* | *ipv6-address* | *peer-group-name*} **send-label**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	hostname <i>name</i> Example: Router(config)# hostname CSC-PE1	Specifies or modifies the host name for the network server.

	Command or Action	Purpose
Step 4	router bgp <i>autonomous-system-number</i> Example: Router(config)# router bgp 100	Configures the BGP routing process.
Step 5	address-family ipv6 [<i>vrf vrf-name</i>] [<i>unicast</i> <i>multicast</i>] Example: Router(config-router)# address-family ipv6 vrf ISP2	Enters address family configuration mode.
Step 6	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } remote-as <i>as-number</i> Example: Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 remote-as 200	Adds an entry to the multiprotocol BGP neighbor table.
Step 7	neighbor { <i>ip-address</i> <i>peer-group-name</i> <i>ipv6-address</i> } activate Example: Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 activate	Enables the exchange of information for this address family with the specified BGP neighbor.
Step 8	neighbor { <i>ip-address</i> <i>ipv6-address</i> <i>peer-group-name</i> } send-label Example: Router(config-router-af)# neighbor FE80::866C:99%Serial0/0 send-label	Enables label exchange for this address family to this neighbor.

Verifying and Troubleshooting IPv6 VPN

When users troubleshoot IPv6, any function that works similarly to VPNv4 will likely work for IPv6, therefore minimizing the learning curve for new IPv6 users. Few of the tools and commands used to troubleshoot 6PE and 6VPE are specific to IPv6; rather, the troubleshooting methodology is the same for both IPv4 and IPv6, and the commands and tools often vary by only one keyword.

The following tasks describe how to verify IPv6 VPN and troubleshoot problems in the specified scenarios:

- [Verifying and Troubleshooting Routing, page 48](#)
- [Verifying and Troubleshooting Forwarding, page 50](#)
- [Debugging Routing and Forwarding, page 54](#)

Verifying and Troubleshooting Routing

Deploying 6PE and 6VPE involves principally BGP. The same set of commands used for VPNv4 can be used (with different set of arguments) for IPv6, and similar outputs are obtained.

The following examples can be used to verify and troubleshoot BGP deployments:

- [BGP IPv6 Activity Summary, page 49](#)
- [Dumping the BGP IPv6 Tables, page 49](#)
- [Dumping the IPv6 Routing Tables, page 49](#)

BGP IPv6 Activity Summary

The following example displays a summary of BGP IPv6 activity:

```
Router# show bgp ipv6 summary

For address family: IPv6 Unicast
BGP router identifier 192.168.2.126, local AS number 33751
BGP table version is 15, main routing table version 15
12 network entries using 1692 bytes of memory
22 path entries using 1672 bytes of memory
5/4 BGP path/bestpath attribute entries using 580 bytes of memory
14 BGP rrinfo entries using 336 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 4328 total bytes of memory
Dampening enabled. 0 history paths, 0 dampened paths
BGP activity 13/1 prefixes, 23/1 paths, scan interval 60 secs
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.2.146   4 33751    991    983     15   0   0 16:26:21      10
192.168.2.147   4 33751    991    983     15   0   0 16:26:22      10
FE80::4FB:44%Serial1/0
                  4 20331    982    987     15   0   0 14:55:52       1
```

Dumping the BGP IPv6 Tables

Each table (for example, BGP IPv6, BGP IPv6 VPN) can be reviewed individually, as shown in the following example:

```
Router# show bgp ipv6 unicast

BGP table version is 15, local router ID is 192.168.2.126
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric      LocPrf Weight Path
* i2001:DB8:100::/48 ::FFFF:192.168.2.101      0       100      0 10000 ?
*>i                ::FFFF:192.168.2.101      0       100      0 10000 ?
* i2001:DB8::1/128  ::FFFF:192.168.2.101      0       100      0 i
*>i                ::FFFF:192.168.2.101      0       100      0 i
```

Dumping the IPv6 Routing Tables

IPv6 routing tables can be displayed, to identify each routing protocol contributor to routable entries, as shown in the following example:

```
Router# show ipv6 route

IPv6 Routing Table - default - 13 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, R - RIP, I1 - ISIS L1, I2 - ISIS L2
       IA - ISIS interarea, IS - ISIS summary
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
B   2001:DB8:100::/48 [200/0]
    via 192.168.2.101%Default-IP-Routing-Table, indirectly connected
```

```

B    2001:DB8::1/128 [200/0]
    via 192.168.2.101%Default-IP-Routing-Table, c
LC   2001:DB8::26/128 [0/0]
    via Loopback0, receive

```

Note that, from an IPv6 routing perspective, entries reachable over the MPLS backbone are listed as being indirectly connected, because MPLS is providing a Layer 2 tunnel mechanism.

Verifying and Troubleshooting Forwarding

Forwarding anomalies should be detected and understood so that users can perform troubleshooting. Commands such as **ping ipv6** and **traceroute ipv6** are used to validate data-plane connectivity and detect traffic black-holing. Commands such as **traceroute mpls** and **show mpls forwarding** can pinpoint a damaged node, interface, and forwarding error correction (FEC). At the edge, troubleshooting forwarding failures for a particular IPv6 destination commonly leads to breaking down the recursive resolution into elementary pieces. This task requires combining analysis of IPv6 routing (iBGP or eBGP), IP routing (IS-IS or OSPF), label distribution (BGP, LDP, or RSVP), and adjacency resolution to find a resolution breakage.

The following examples describe how to verify IPv6 VPN and troubleshoot various IPv6 VPN forwarding situations:

- [PE-CE Connectivity, page 50](#)
- [PE Imposition Path, page 51](#)
- [PE Disposition Path, page 52](#)
- [Label Switch Path, page 53](#)

PE-CE Connectivity

The **ipv6 ping** and **traceroute** commands are useful to check connectivity from a PE to a CE, whether locally attached or remote over the MPLS backbone.

When a router is locally attached, one can use the **ipv6 ping** command with the CE link-local address (used for eBGP peering), as shown in the following example:

```

Router# ping FE80::4F6B:44%Serial1/0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::4F6B:44, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms

```

The **ipv6 ping** command also can be used to test remote PE or CE reachability, but only IPv6 global addresses can be used (link-local addresses are not advertised beyond the link):

```

Router# ping 2001:DB8:1120:1::44

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:DB8:1120:1:44::1, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/33/48 ms

```

Note that the **ping ipv6** and **traceroute** command functions over MPLS require PEs and CEs to announce one IPv6 global prefix. Each 6PE router announces 2001:DB8::PE#/128, filtered at the autonomous system edge. Each IPv6 CE configures 2001:DB8:prefix:CE#/128 and announces it as part as its less-specific prefix (2001:DB8:prefix::/n).

Reachability of remote PEs and CEs can be tested by using the **traceroute** command. If you have configured all PEs with the **no mpls ip propagate-ttl forwarded** command, when the **traceroute** command is executed from a CE, its output will show only the IPv6 nodes:

```
Router# traceroute 2001:DB8::1

Type escape sequence to abort.
Tracing the route to 2001:DB8::1
 1 2001:DB8::26 [AS 33751] 32 msec 32 msec 20 msec
 2 2001:DB8::1 [AS 33751] [MPLS: Label 73 Exp 0] 20 msec 20 msec 20 msec
 3 2001:DB8::1 [AS 33751] 28 msec 20 msec 20 msec
```

After the P routers have been upgraded with images that support ICMPv6, the **traceroute** command executed on the PE router (Time to Live [TTL] is then propagated) will also show P routers' responses, as shown in the following example:

```
Router# traceroute 2001:DB8::1

Type escape sequence to abort.
Tracing the route to 2001:DB8::1
 1 ::FFFF:172.20.25.1 [MPLS: Labels 38/73 Exp 0] 40 msec 32 msec 32 msec
 2 ::FFFF:172.20.10.1 [MPLS: Labels 30/73 Exp 0] 60 msec 32 msec 32 msec
 3 2001:DB8::1 [MPLS: Label 73 Exp 0] 32 msec 32 msec 16 msec
```

When run from a 6VPE router, both the **ping ipv6** and **traceroute** commands accept a *vrf* argument, exactly as in the case of VPNv4.

Note that the **traceroute** command is useful for evaluating the path across the MPLS backbone, but not for troubleshooting data-plane failures. The P routers are IPv6 unaware (and are also VPNv4 unaware), so the ICMPv6 messages that they generate in response to the **traceroute** command are forwarded to the egress PE using the received label stack. The egress PE can route the ICMPv6 message to the source of the traceroute. When the MPLS path is broken, it is also broken from the ICMP message, which cannot reach the egress PE.

PE Imposition Path

On Cisco routers, the most useful tool for troubleshooting the imposition path for IPv6 is the **show ipv6 cef** command.

Dumping IPv6 Forwarding Table

You can use the **show ipv6 cef** command to display the forwarding table with label stacks used for each destination prefix, as shown in the following example:

```
Router# show ipv6 cef

2001:DB8:100::/48
  nexthop 172.20.25.1 Serial0/0 label 38 72
2001:DB8::1/128
  nexthop 172.20.25.1 Serial0/0 label 38 73
2001:DB8::26/128
  attached to Loopback0, receive
```

Details of an IPv6 Entry in the Forwarding Table

You can use the **show ipv6 cef** command to display details for a specific entry and to analyze how the destination was resolved and the label stack computed, as shown in the following example:

```
Router# show ipv6 cef 2001:DB8:100::/48 internal

2001:DB8:100::/48, epoch 0, RIB[B], refcount 4
  sources: RIB
```

```
..
recursive via 192.168.2.101[IPv4:Default] label 72, fib 0252B1F8, 1 terminal fib
  path 024F56A8, path list 024F0BA8, share 0/1, type attached nexthop
  ifnums: (none)
  path_list contains at least one resolved destination(s). HW IPv4 notified.
  nexthop 172.20.25.1 Serial0/0 label 38, adjacency IP adj out of Serial0/0 0289BEF0
  output chain: label 72 label 38 TAG adj out of Serial0/0 0289BD80
```

Details of a BGP Entry in the BGP Table

The detailed output in the previous example shows that each label composing the label stack has a different origin that can be tracked down individually. The BGP table has the bottom label, as shown in the following example:

```
Router# show bgp ipv6 unicast 2001:DB8:100::/48
```

```
BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    1
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.147 (192.168.2.147)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Originator: 192.168.2.101, Cluster list: 192.168.2.147,
      mpls labels in/out nolabel/72
  10000
    ::FFFF:192.168.2.101 (metric 30) from 192.168.2.146 (192.168.2.146)
      Origin incomplete, metric 0, localpref 100, valid, internal, best
      Originator: 192.168.2.101, Cluster list: 192.168.2.146,
      mpls labels in/out nolabel/72
```

LDP, as shown in this example, displays the other labels:

```
Router# show mpls ldp bindings 192.168.2.101 32
```

```
lib entry: 192.168.2.101/32, rev 56
  local binding: label: 40
  remote binding: lsr: 192.168.2.119:0, label: 38
```

```
Router# show mpls ldp bindings 172.20.25.0 24
```

```
lib entry: 172.20.25.0/24, rev 2
  local binding: label: imp-null
  remote binding: lsr: 192.168.2.119:0, label: imp-null
```

PE Disposition Path

Use the following examples to troubleshoot the disposition path.

- [Dumping the MPLS Forwarding Table, page 52](#)
- [BGP Label Analysis, page 53](#)

Dumping the MPLS Forwarding Table

The following example illustrates MPLS forwarding table information for troubleshooting the disposition path.

```
Router# show mpls forwarding-table
```

Local Label	Outgoing Label or VC	Prefix or Tunnel Id	Bytes Label Switched	Outgoing interface	Next Hop
16	Pop Label	192.168.2.114/32	0	Se0/0	point2point
17	26	192.168.2.146/32	0	Se0/0	point2point

```

..
72   No Label      2001:DB8:100::/48      63121      Se1/0      point2point
73   Aggregate    2001:DB8::1/128      24123

```

BGP Label Analysis

The following example illustrates the label used for switching, which has been announced by iBGP (6PE in this example) and can be checked:

```
Router# show bgp ipv6 2001:DB8:100::/48
```

```

BGP routing table entry for 2001:DB8:100::/48, version 2
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  10000
    FE80::2710:2 (FE80::2710:2) from FE80::2710:2%Serial1/0 (192.168.2.103)
      Origin incomplete, metric 0, localpref 100, valid, external, best,

```

Label Switch Path

Because the 6PE and 6VPE LSP endpoints are IPv4 addresses, the IPv4 tools for troubleshooting LSPs are useful for detecting data-plane failures that would lead to IPv6 traffic black-holing.

Analyzing the Label Switch Path

The following example displays the LSP IPv4 end:

```
Router# show ipv6 route 2001:DB8::1/128
```

```

Routing entry for 2001:DB8::1/128
  Known via "bgp 33751", distance 200, metric 0, type internal
  Route count is 1/1, share count 0
  Routing paths:
    192.168.2.101%Default-IP-Routing-Table indirectly connected
    MPLS Required
    Last updated 02:42:12 ago

```

Traceroute LSP Example

The following example shows the traceroute LSP:

```
Router# traceroute mpls ipv4 192.168.2.101/32 verbose
```

```

Tracing MPLS Label Switched Path to 192.168.2.101/32, timeout is 2 seconds
Codes: '!' - success, 'Q' - request not transmitted,
       '.' - timeout, 'U' - unreachable,
       'R' - downstream router but not target,
       'M' - malformed request
Type escape sequence to abort.
 0 172.20.25.2 0.0.0.0 MRU 1500 [Labels: 38 Exp: 0]
R 1 172.20.25.1 0.0.0.0 MRU 1500 [Labels: 30 Exp: 0] 40 ms, ret code 6
R 2 172.20.10.1 0.0.0.0 MRU 1504 [Labels: implicit-null Exp: 0] 60 ms, ret code 6
! 3 172.20.40.1 48 ms

```

VRF Information

The following entries show VRF information for 6VPE.

show ipv6 cef vrf

The following is sample output from a Cisco Express Forwarding FIB associated with a VRF named cisco1:

```
Router# show ipv6 cef vrf cisco1

2001:8::/64
  attached to FastEthernet0/0
2001:8::3/128
  receive
2002:8::/64
  nexthop 10.1.1.2 POS4/0 label 22 19
2010::/64
  nexthop 2001:8::1 FastEthernet0/0
2012::/64
  attached to Loopback1
2012::1/128
  receive
```

show ipv6 route vrf

The following is sample output regarding an IPv6 routing table associated with a VRF named cisco1:

```
Router# show ipv6 route vrf cisco1

IPv6 Routing Table cisco1 - 6 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
C   2001:8::/64 [0/0]
    via ::, FastEthernet0/0
L   2001:8::3/128 [0/0]
    via ::, FastEthernet0/0
B   2002:8::/64 [200/0]
    via ::FFFF:192.168.1.4,
B   2010::/64 [20/1]
    via 2001:8::1,
C   2012::/64 [0/0]
    via ::, Loopback1
L   2012::1/128 [0/0]
    via ::, Loopback1
```

Debugging Routing and Forwarding

For troubleshooting of routing and forwarding anomalies, enabling debugging commands can prove useful, although several debug messages can slow the router and harm the usability of such a tool. For this reason, use **debug** commands with caution. The **debug ipv6 cef**, **debug mpls packet**, and **debug ipv6 packet** commands are useful for troubleshooting the forwarding path; the **debug bgp ipv6** and **debug bgp vpnv6** commands are useful for troubleshooting the control plane.

Configuration Examples for Implementing IPv6 VPN over MPLS

This section provides the following configuration example:

- [IPv6 VPN Configuration Using IPv4 Next Hop: Example, page 54](#)

IPv6 VPN Configuration Using IPv4 Next Hop: Example

The following example illustrates a 6VPE next hop:

```
interface Loopback0
```

```

ip address 192.168.2.11 255.255.255.255
!
router bgp 100
 neighbor 192.168.2.10 remote-as 100
 neighbor 192.168.2.10 update-source Loopback0
!
 address-family vpnv6
  neighbor 192.168.2.10 activate
  neighbor 192.168.2.10 send-community extended
 exit-address-family

```

By default, the next hop advertised will be the IPv6 VPN address:

```
[0:0]::FFFF:192.168.2.10
```

Note that it is a 192-bit address in the format of `[RD]::FFFF:IPv4-address`.

When the BGP IPv6 VPN peers share a common subnet, the MP_REACH_NLRI attribute contains a link-local address next hop in addition to the global address next hop. This situation typically occurs in an interautonomous-system topology when ASBRs are facing each other. In that case, the link-local next hop is used locally, and the global next hop is readvertised by BGP.

The BGP next hop is the keystone for building the label stack. The inner label is obtained from the BGP NLRI, and the outer label is the label distribution protocol (LDP) label to reach the IPv4 address embedded into the BGP next hop.

Additional References

The following sections provide references related to the Implementing IPv6 VPN over MPLS feature.

Related Documents

Related Topic	Document Title
IPv6 commands	Cisco IOS IPv6 Command Reference

Standards

Standard	Title
draft-bonica-internet-icmp	<i>ICMP Extensions for Multiprotocol Label Switching</i>
draft-ietf-idr-bgp-ext-communities-0x.txt	<i>Cooperative Route Filtering Capability for BGP-4</i>

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 1267	<i>A Border Gateway Protocol 3 (BGP-3)</i>
RFC 1772	<i>Application of the Border Gateway Protocol in the Internet</i>
RFC 1918	<i>Address Allocation for Private Internets</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 3107	<i>Carrying Label Information in BGP-4</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 3513	<i>Internet Protocol Version 6 (IPv6) Addressing Architecture</i>
RFC 4007	<i>IPv6 Scoped Address Architecture</i>
RFC 4193	<i>Unique Local IPv6 Unicast Addresses</i>
RFC 4364	<i>BGP MPLS/IP Virtual Private Networks (VPNs)</i>
RFC 4382	<i>MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base</i>
RFC 4659	<i>BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **address-family ipv4 (BGP)**

- **address-family ipv6**
- **address-family vpnv6**
- **aggregate-address**
- **bgp log-neighbor-changes**
- **clear bgp ipv6**
- **context**
- **debug bgp vpnv6 unicast**
- **hostname**
- **ip address**
- **ip route-cache**
- **ip router isis**
- **match mpls-label**
- **maximum routes**
- **mpls ldp router-id**
- **mpls traffic-eng auto-bw timers**
- **neighbor ebgp-multihop**
- **neighbor next-hop-unchanged**
- **neighbor remote-as**
- **neighbor route-map**
- **neighbor route-reflector-client**
- **neighbor send-community**
- **network (IPv6)**
- **passive-interface (IPv6)**
- **ping vrf**
- **rd**
- **router bgp**
- **route-target**
- **service pad**
- **service password-encryption**
- **service timestamps**
- **set extcommunity**
- **set mpls-label**
- **show bgp vpnv6 unicast**
- **show ipv6 cef vrf**
- **show ipv6 route vrf**
- **show monitor event-trace vpn-mapper**
- **show platform software vpn**
- **show vrf**

- **vpn**
- **vrf definition**
- **vrf forwarding**

Feature Information for Implementing IPv6 VPN over MPLS

Table 1 lists the release history for this feature.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing IPv6 VPN over MPLS

Feature Name	Releases	Feature Information
IPv6 VPN over MPLS (6VPE)	12.2(28)SB 12.2(33)SRB 12.2(33)SXI 12.4(20)T	The IPv6 VPN (6VPE) over a MPLS IPv4 core infrastructure feature allows ISPs to offer IPv6 VPN services to their customers. This entire document provides information about this feature.
MPLS VPN 6VPE support over IP tunnels	12.2(33)SRB1 12.2(33)SXI	This feature allows the use of IPv4 GRE tunnels to provide IPv6 VPN over MPLS functionality to reach the BGP next hop. This following sections provide information about this feature: <ul style="list-style-type: none"> • MPLS Forwarding, page 5

Glossary

- **6VPE router**—Provider edge router providing BGP-MPLS IPv6 VPN service over an IPv4-based MPLS core. It is a IPv6 VPN PE, dual-stack router that implements 6PE concepts on the core-facing interfaces.
- **customer edge (CE) router**—A service provider router that connects to VPN customer sites.
- **Forwarding Information Base (FIB)**—Table containing the information necessary to forward IP datagrams. At a minimum, the FIB contains the interface identifier and next-hop information for each reachable destination network prefix.
- **inbound route filtering (IRF)**—A BGP capability used for filtering incoming BGP updates that are not to be imported by the receiving PE router.
- **IPv6 provider edge router (6PE router)**—Router running a BGP-based mechanism to interconnect IPv6 islands over an MPLS-enabled IPv4 cloud.
- **IPv6 VPN address**—A IPv6 VPN address is a 24-byte identifier, beginning with an 8-byte route distinguisher (RD) and ending with a 16-byte IPv6 address. Sometimes it is called an IPv6 VPN address.
- **IPv6 VPN address family**—The address-family identifier (AFI) identifies a particular network-layer protocol and the subsequent AFI (SAFI) provides additional information. The AFI IPv6 SAFI VPN (AFI=2, SAFI=128) is called the IPv6 VPN address family. Sometimes it is called the IPv6 VPN address family. Similarly AFI IPv4 SAFI VPN is the VPNv4 address family.
- **network layer reachability information (NLRI)**—BGP sends routing update messages containing NLRI to describe a route and how to get there. In this context, an NLRI is a prefix. A BGP update message carries one or more NLRI prefixes and the attributes of a route for the NLRI prefixes; the route attributes include a BGP next hop gateway address and community values.
- **outbound route filtering (ORF)**—A BGP capability used to filtering outgoing BGP routing updates.
- **point of presence (POP)**—Physical location where an interexchange carrier installed equipment to interconnect with a local exchange carrier.
- **provider edge (PE) router**—A service provider router connected to VPN customer sites.
- **route distinguisher (RD)**—A 64-bit value prepended to an IPv6 prefix to create a globally unique IPv6 VPN address.
- **Routing Information Base (RIB)**—Also called the routing table.
- **Virtual routing and forwarding (VRF)**—A VPN routing and forwarding instance in a PE.
- **VRF table**—A routing and a forwarding table associated to a VRF. This is a customer-specific table that enables the PE router to maintain independent routing states for each customer.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2007–2009 Cisco Systems, Inc. All rights reserved.



Implementing Policy-Based Routing for IPv6

First Published: March 1, 2004

Last Updated: July 25, 2005

This module describes policy-based routing (PBR) for IPv6. PBR in both IPv6 and IPv4 allows a user to manually configure how received packets should be routed. PBR allows the user to identify packets using several attributes and to specify the next hop or output interface to which the packet should be sent. PBR also provides a basic packet-marking capability.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Policy-Based Routing for IPv6”](#) section on [page 13](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing Policy-Based Routing for IPv6, page 2](#)
- [Restrictions for Implementing Policy-Based Routing for IPv6, page 2](#)
- [Information About Implementing Policy-Based Routing for IPv6, page 2](#)
- [How to Implement Policy-Based Routing for IPv6, page 4](#)
- [Configuration Examples for Implementing Policy-Based Routing for IPv6, page 9](#)
- [Additional References, page 10](#)
- [Command Reference, page 12](#)
- [Feature Information for Implementing Policy-Based Routing for IPv6, page 13](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2004–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing Policy-Based Routing for IPv6

- This module assumes that you are familiar with *IPv6 addressing and basic configuration*. Refer to [Implementing IPv6 Addressing and Basic Connectivity](#) for more information.
- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for IPv4 configuration and command reference information, as needed.

Restrictions for Implementing Policy-Based Routing for IPv6

Distributed Cisco Express Forwarding is supported on the Cisco 7500 series routers only.

Information About Implementing Policy-Based Routing for IPv6

To configure PBR for IPv6 for Cisco IOS software, you should understand the following concepts:

- [Policy-Based Routing Overview, page 2](#)
- [How Policy-Based Routing Works, page 3](#)
- [When to Use Policy-Based Routing, page 4](#)

Policy-Based Routing Overview

PBR gives you a flexible means of routing packets by allowing you to configure a defined policy for traffic flows, which lessens reliance on routes derived from routing protocols. To this end, PBR gives you more control over routing by extending and complementing the existing mechanisms provided by routing protocols. PBR allows you to set the IPv6 precedence. It also allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.

PBR for IPv6 may be applied to both forwarded and originated IPv6 packets. For forwarded packets, PBR for IPv6 will be implemented as an IPv6 input interface feature, supported in the process, Cisco Express Forwarding, and distributed Cisco Express Forwarding forwarding paths.

You can set up PBR as a way to route packets based on configured policies. For example, you can implement routing policies to allow or deny paths based on the identity of a particular end system, an application protocol, or the size of packets.

PBR allows you to perform the following tasks:

- Classify traffic based on extended access list criteria. Access lists, then, establish the match criteria.
- Set IPv6 precedence bits, giving the network the ability to enable differentiated classes of service.
- Route packets to specific traffic-engineered paths; you might need to route them to allow a specific quality of service (QoS) through the network.

Policies can be based on IPv6 address, port numbers, protocols, or size of packets. For a simple policy, you can use any one of these descriptors; for a complex policy, you can use all of them.

PBR allows you to classify and mark packets at the edge of the network. PBR marks a packet by setting its precedence value. The precedence value can be used directly by routers in the network core to apply the appropriate QoS to a packet, which keeps packet classification at your network edge.

How Policy-Based Routing Works

All packets received on an interface with PBR enabled are passed through enhanced packet filters known as route maps. The route maps used by PBR dictate the policy, determining where to forward packets.

Route maps are composed of statements. The route map statements can be marked as permit or deny, and they are interpreted in the following ways:

- If a packet matches all match statements for a route map that is marked as permit, then the router attempts to policy route the packet using the set statements. Otherwise, the packet is forwarded normally.
- If the packet matches any match statements for a route map that is marked as deny, then the packet is not subject to PBR and is forwarded normally.
- If the statement is marked as permit and the packets do not match any route map statements, the packets are sent back through the normal forwarding channels and destination-based routing is performed.

You specify PBR on the interface that receives the packet, not on the interface from which the packet is sent.

Packet Matching

PBR for IPv6 will match packets using the **match ipv6 address** command in the associated PBR route map. Packet match criteria are those criteria supported by IPv6 access lists, as follows:

- Input interface
- Source IPv6 address (using a prefix list or a standard or extended access list [ACL])
- Destination IPv6 address (standard or extended ACL)
- Protocol (extended ACL)
- Source port and destination port (extended ACL)
- Differentiated services code point (DSCP) (extended ACL)
- Flow-label (extended ACL)
- Fragment (extended ACL)

Packets may also be matched by length using the match length statement in the PBR route map.

Match statements are evaluated first by the criteria specified in the **match ipv6 address** command and then by criteria specified in the **match length** command. Therefore, if both an ACL and a length statement are used, a packet will first be subject to an ACL match. Only packets that pass the ACL match will then be subject to the length match. Finally, only packets that pass both the ACL and the length statement will be policy routed.

Packet Forwarding Using Set Statements

PBR for IPv6 packet forwarding is controlled using a number of set statements in the PBR route map. These set statements are evaluated individually in the order shown, and PBR will attempt to forward the packet using each of the of the set statements in turn. PBR evaluates each set statement by itself, without reference to any prior or subsequent set statement.

You may set multiple forwarding statements in the PBR for IPv6 route map. The following set statements may be specified:

- IPv6 next hop. The next hop to which the packet should be sent. The next hop must be present in the Routing Information Base (RIB), it must be directly connected, and it must be a global IPv6 address. If the next hop is invalid, the set statement is ignored.
- Output interface. A packet is forwarded out of a specified interface. An entry for the packet destination address must exist in the IPv6 RIB, and the specified output interface must be in the path set. If the interface is invalid, the statement is ignored.
- Default IPv6 next hop. The next hop to which the packet should be sent. It must be a global IPv6 address. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.
- Default output interface. The packet is forwarded out a specified interface. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.

**Note**

The order in which PBR evaluates the set statements is the order in which they are listed above. This order may differ from the order in which route-map set statements are listed by Cisco IOS **show** commands.

When to Use Policy-Based Routing

You might use PBR if you want certain packets to be routed some way other than the obvious shortest path. For example, PBR can be used to provide the following functionality:

- Equal access
- Protocol-sensitive routing
- Source-sensitive routing
- Routing based on interactive versus batch traffic
- Routing based on dedicated links

Some applications or traffic can benefit from QoS-specific routing; for example, you could transfer stock records to a corporate office on a higher-bandwidth, higher-cost link for a short time while sending routine application data such as e-mail over a lower-bandwidth, lower-cost link.

How to Implement Policy-Based Routing for IPv6

The tasks in the following sections explain how to implement policy-based routing for IPv6:

- [Enabling PBR on an Interface, page 4](#)
- [Enabling Local PBR for IPv6, page 7](#)
- [Enabling Cisco Express Forwarding-Switched PBR for IPv6, page 7](#)
- [Troubleshooting PBR for IPv6, page 8](#)

Enabling PBR on an Interface

To enable PBR for IPv6, you must create a route map that specifies the packet match criteria and desired policy-route action. Then you associate the route map on the required interface. All packets arriving on the specified interface that match the match clauses will be subject to PBR.

This task enables PBR on an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
4. **match length** *minimum-length maximum-length*
or
match ipv6 address {**prefix-list** *prefix-list-name* | *access-list-name*}
5. **set ipv6 precedence** *precedence-value*
or
set ipv6 next-hop *global-ipv6-address [global-ipv6-address...]*
or
set interface *type number [...type number]*
or
set ipv6 default next-hop *global-ipv6-address [global-ipv6-address...]*
or
set default interface *type number [...type number]*
6. **exit**
7. **interface** *type number*
8. **ipv6 policy route-map** *route-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	route-map <i>map-tag</i> [permit deny] [<i>sequence-number</i>] Example: Router(config)# route-map rip-to-ospf permit	Defines the conditions for redistributing routes from one routing protocol into another, or enables policy routing. <ul style="list-style-type: none">Use the route-map command to enter route-map configuration mode.

	Command or Action	Purpose
Step 4	<p>match length <i>minimum-length maximum-length</i> or</p> <p>match ipv6 address {prefix-list <i>prefix-list-name</i> <i>access-list-name</i>}</p> <p>Example: Router(config-route-map)# match length 3 200 or Router(config-route-map)# match ipv6 address marketing</p>	<p>Specifies the match criteria.</p> <ul style="list-style-type: none"> You can specify any or all of the following: <ul style="list-style-type: none"> Matches the Level 3 length of the packet. Matches a specified IPv6 access list. If you do not specify a match command, the route map applies to all packets.
Step 5	<p>set ipv6 precedence <i>precedence-value</i> or</p> <p>set ipv6 next-hop <i>global-ipv6-address</i> [<i>global-ipv6-address...</i>] or</p> <p>set interface <i>type number</i> [...<i>type number</i>] or</p> <p>set ipv6 default next-hop <i>global-ipv6-address</i> [<i>global-ipv6-address...</i>] or</p> <p>set default interface <i>type number</i> [...<i>type number</i>]</p> <p>Example: Router(config-route-map)# set ipv6 precedence 1 or Router(config-route-map)# set ipv6 next-hop 2001:0db8:2003:1::95 or Router(config-route-map)# set interface ethernet 0 or Router(config-route-map)# set ipv6 default next-hop 2001:0db8:2003:1::95 or Router(config-route-map)# set default interface ethernet 0</p>	<p>Specifies the action or actions to take on the packets that match the criteria.</p> <ul style="list-style-type: none"> You can specify any or all of the following: <ul style="list-style-type: none"> Sets precedence value in the IPv6 header. Sets next hop to which to route the packet (the next hop must be adjacent). Sets output interface for the packet. Sets next hop to which to route the packet, if there is no explicit route for this destination. Sets output interface for the packet, if there is no explicit route for this destination.
Step 6	<p>exit</p> <p>Example: Router(config-route-map)# exit</p>	<p>Returns the router to global configuration mode.</p>

	Command or Action	Purpose
Step 7	interface <i>type number</i> Example: Router(config)# interface FastEthernet 1/0	Specifies an interface type and number, and places the router in interface configuration mode.
Step 8	ipv6 policy route-map <i>route-map-name</i> Example: Router(config-if)# ipv6 policy-route-map interactive	Identifies a route map to use for IPv6 PBR on an interface.

Enabling Local PBR for IPv6

Packets that are generated by the router are not normally policy routed. This task enables local PBR for IPv6 for such packets, indicating which route map the router should use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 local policy route-map** *route-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 local policy route-map <i>route-map-name</i> Example: Router(config)# ipv6 local policy route-map pbr-src-90	Configures PBR for IPv6 for packets generated by the router.

Enabling Cisco Express Forwarding-Switched PBR for IPv6

Beginning in Cisco IOS Release 12.3(7)T, PBR for IPv6 is supported in the Cisco Express Forwarding switching path. Cisco Express Forwarding-switched PBR is the optimal way to perform PBR on a router.

No special configuration is required to enable Cisco Express Forwarding-switched PBR for IPv6. It is on by default as soon as you enable Cisco Express Forwarding and PBR on the router.

Verifying Configuration and Operation of PBR for IPv6

This task explains how to display information to verify the configuration and operation of PBR for IPv6.

SUMMARY STEPS

1. **enable**
2. **show ipv6 policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	show ipv6 policy Example: Router# show ipv6 policy	Displays IPv6 policy routing packet activity.

Troubleshooting PBR for IPv6

Policy routing looks at various parts of the packet and then routes the packet based on certain user-defined attributes in the packet. This task helps you determine what policy routing is following, whether a packet matches the criteria, and if so, the resulting routing information for the packet.

SUMMARY STEPS

1. **enable**
2. **debug ipv6 policy** [*access-list-name*]
3. **show route-map** [*map-name* | **dynamic** [*dynamic-map-name* | **application** [*application-name*]] | **all**] [**detailed**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	debug ipv6 policy [access-list-name]	Displays IPv6 policy routing packet activity.
	Example: Router# debug ipv6 policy	
Step 3	show route-map [map-name dynamic [dynamic-map-name application [application-name]] all] [detailed]	Displays all route maps configured or only the one specified.
	Example: Router# show route-map	

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 policy Command, page 9](#)
- [Sample Output from the show route-map Command, page 9](#)

Sample Output from the show ipv6 policy Command

The **show ipv6 policy** command displays PBR configuration, as shown in the following example:

```
Router# show ipv6 policy
```

```
Interface          Routemap
Ethernet0/0        src-1
```

Sample Output from the show route-map Command

The **show route-map** command displays specific route-map information, such as a count of policy matches:

```
Router# show route-map
```

```
route-map bill, permit, sequence 10
Match clauses:
Set clauses:
Policy routing matches:0 packets, 0 bytes
```

Configuration Examples for Implementing Policy-Based Routing for IPv6

The following sections provide PBR for IPv6 configuration examples:

- [Enabling PBR on an Interface: Example, page 10](#)
- [Enabling Local PBR for IPv6: Example, page 10](#)

Enabling PBR on an Interface: Example

In the following example, a route map named pbr-dest-1 is created and configured, specifying packet match criteria and desired policy-route action. Then, PBR is enabled on Ethernet interface 0/0.

```
ipv6 access-list match-dest-1
  permit ipv6 any 2001:0db8:2001:1760::/32

route-map pbr-dest-1 permit 10
  match ipv6 address match-dest-1
  set interface Ethernet 0/0

interface Ethernet0/0
  ipv6 policy-route-map interactive
```

Enabling Local PBR for IPv6: Example

In the following example, packets with a destination IPv6 address matching that allowed by access list pbr-src-90 are sent to the router at IPv6 address 2001:0db8:2003:1::95:

```
ipv6 access-list src-90
  permit ipv6 host 2001:0db8:2003::90 2001:0db8:2001:1000::/64

route-map pbr-src-90 permit 10
  match ipv6 address src-90
  set ipv6 next-hop 2001:0db8:2003:1::95

ipv6 local policy route-map pbr-src-90
```

Additional References

The following sections provide references related to the implementing policy-based routing for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 addressing and basic configuration	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
QoS for IPv6	“Implementing QoS for IPv6,” Cisco IOS IPv6 Configuration Guide
Multicast Border Gateway Protocol (BGP) for IPv6	“Implementing Multiprotocol BGP for IPv6,” Cisco IOS IPv6 Configuration Guide
Access control lists for IPv6	“Implementing Traffic Filters and Firewalls for IPv6 Security,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide

Related Topic	Document Title
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference
IPv4 Quality of Service	“ Quality of Service Overview ,” <i>Cisco IOS Quality of Service Solutions Configuration Guide</i>

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **debug ipv6 policy**
- **ipv6 local policy route-map**
- **ipv6 policy route-map**
- **match ipv6 address**
- **match length**
- **route-map**
- **set default interface**
- **set interface**
- **set ipv6 default next-hop**
- **set ipv6 next-hop (PBR)**
- **set ipv6 precedence**
- **show ipv6 policy**

Feature Information for Implementing Policy-Based Routing for IPv6

Table 17 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.3(7)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 17 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 17 Feature Information for Mobile IPv6

Feature Name	Releases	Feature Information
IPv6 routing: IPv6 policy-based routing	12.2(30)S 12.3(7)T 12.4 12.4(2)T	Policy-based routing for IPv6 in Cisco IOS software allows a user to manually configure how received packets should be routed. This entire document describes this feature.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2003–2009 Cisco Systems, Inc. All rights reserved.



Implementing QoS for IPv6

First Published: November 25, 2002

Last Updated: October 5, 2008

This module provides information about and tasks for implementing quality of service (QoS) features in IPv6 environments, specifically the application of the Differentiated Services (DiffServ) QoS features to IPv6 packets.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing QoS for IPv6”](#) section on page 19.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing QoS for IPv6, page 2](#)
- [Restrictions for Implementing QoS for IPv6, page 2](#)
- [Information About Implementing QoS for IPv6, page 2](#)
- [How to Implement QoS for IPv6, page 4](#)
- [Configuration Examples for Implementing QoS for IPv6, page 15](#)
- [Additional References, page 16](#)
- [Command Reference, page 17](#)
- [Feature Information for Implementing QoS for IPv6, page 19](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing QoS for IPv6

This document assumes that you are familiar with IPv4. Refer to the publications referenced in the [“Additional References”](#) section for IPv4 configuration and command reference information.

Restrictions for Implementing QoS for IPv6

The following QoS features are not supported for managing IPv6 traffic:

- Compressed Real-Time Protocol (CRTTP)
- Network-based application recognition (NBAR)
- Committed access rate (CAR)
- Priority queueing (PQ)
- Custom queueing (CQ)

Platform-Specific Information and Restrictions

IPv6 QoS is supported on Cisco 12000 series Internet routers in Cisco IOS Release 12.0(28)S. Certain features of IPv6 QoS are not supported in Release 12.0(28)S. These features include packet classification.

Information About Implementing QoS for IPv6

The following sections provide information about the QoS features available for managing IPv6 traffic:

- [Implementation Strategy for QoS for IPv6, page 2](#)
- [Packet Classification in IPv6, page 3](#)
- [Policies and Class-Based Packet Marking in IPv6 Networks, page 3](#)
- [Congestion Management in IPv6 Networks, page 3](#)
- [Congestion Avoidance for IPv6 Traffic, page 4](#)
- [Traffic Policing in IPv6 Environments, page 4](#)

Implementation Strategy for QoS for IPv6

IPv6 packets are forwarded by paths that are different from those for IPv4. QoS features supported for IPv6 environments include packet classification, queueing, traffic shaping, weighted random early detection (WRED), class-based packet marking, and policing of IPv6 packets. These features are available at both the process switching and Cisco Express Forwarding switching paths of IPv6.

All of the QoS features available for IPv6 environments are managed from the modular QoS command-line interface (CLI). The modular QoS CLI allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach those traffic policies to interfaces.

To implement QoS in networks running IPv6, follow the same steps that you would follow to implement QoS in networks running only IPv4. At a very high level, the basic steps for implementing QoS are as follows:

1. Know which applications in your network need QoS.

2. Understand the characteristics of the applications so that you can make decisions about which QoS features would be appropriate.
3. Know your network topology so that you know how link layer header sizes are affected by changes and forwarding.
4. Create classes based on the criteria you establish for your network. In particular, if the same network is also carrying IPv4 traffic along with IPv6, decide if you want to treat both of them the same way or treat them separately and specify match criteria accordingly. If you want to treat them the same, use match statements such as **match precedence**, **match dscp**, **set precedence**, and **set dscp**. If you want to treat them separately, add match criteria such as **match protocol ip** and **match protocol ipv6** in a match-all class map.
5. Create a policy to mark each class.
6. Work from the edge toward the core in applying QoS features.
7. Build the policy to treat the traffic.
8. Apply the policy.

Packet Classification in IPv6

Packet classification is available with both process and Cisco Express Forwarding switching path. Classification can be based on IPv6 precedence, differentiated services control point (DSCP), and other IPv6 protocol-specific values that can be specified in IPv6 access lists in addition to other non-IPv6 protocol specific values such as COS, packet length, and QoS group. Once you determine which applications need QoS, you can create classes based on the characteristics of the applications. You can use a variety of match criteria to classify traffic. You can combine various match criteria to segregate, isolate, and differentiate traffic.

The enhancements to the modular QoS CLI (MQC) allow you to create matches on precedence, DSCP, and IPv6 access group values in both IPv4 and IPv6 packets. The **match** command allows matches to be made on DSCP values and precedence for both IPv4 and IPv6 packets. See [“Using the Match Criteria to Manage IPv6 Traffic Flows” section on page 6](#) for configuration guidelines and to see the **match dscp** and **match precedence** command descriptions.

Policies and Class-Based Packet Marking in IPv6 Networks

You can create a policy to mark each class of traffic with appropriate priority values, using either DSCP or precedence. Class-based marking allows you to set the IPv6 precedence and DSCP values for traffic management. The traffic is marked as it enters the router on the ingress interface. The markings are used to treat the traffic (forward, queue) as it leaves the router on the egress interface. Always mark and treat the traffic as close as possible to its source.

Use the **set dscp** and **set precedence** commands for packet marking. These commands have been modified to handle both IPv4 and IPv6 traffic. See the [“Specifying Marking Criteria for IPv6 Packets” section on page 5](#) for configuration guidelines for using these commands.

Congestion Management in IPv6 Networks

Once you have marked the traffic, you can use the markings to build a policy and classify traffic on the rest of the network segments. If you keep the policy simple (no more than about four classes), it will be easier to manage. Class-based and flow-based queueing are supported for IPv6. The processes and tasks

use the same commands and arguments to configure various queueing options for both IP and IPv6. Refer to the [Cisco IOS Quality of Service Solutions Configuration Guide](#) for configuration and usage instructions of queueing features.

Congestion Avoidance for IPv6 Traffic

WRED implements the RED-based drop policy on the packets that are likely to overflow the limits of Class-based Weighted Fair Queueing (CBWFQ). WRED supports class-based and flow-based (using DSCP or precedence values) queueing. The WRED commands apply to both IPv4 and IPv6 with no changes.

Traffic Policing in IPv6 Environments

Congestion management for IPv6 is similar to its implementation for IP packets, and the commands used to configure queueing and traffic shaping features for IPv6 environments are the same commands as those used for IP. Traffic shaping allows you to limit the packet dequeue rate by holding additional packets in the queues and forwarding them as specified by parameters configured for traffic shaping features. Traffic shaping uses flow-based queueing by default. CBWFQ can be used to classify and prioritize the packets. Class-Based Policer and Generic Traffic Shaping (GTS) or Frame Relay Traffic Shaping (FRTS) can be used for conditioning and policing traffic.

Although no changes to existing configuration or command usage for policing are required for use in IPv6 environments, the **police** command has been enhanced to mark both IPv4 and IPv6 packets when the following keyword options are used in confirm action, exceed action, and violate action:

- **set-dscp-transmit**
- **set-precedence-transmit**

How to Implement QoS for IPv6

These configuration tasks describe how to classify traffic with match criteria and use the match criteria to manage traffic flows. The following sections are included:

- [Restrictions for Classifying Traffic in IPv6 Networks, page 4](#) (required)
- [Specifying Marking Criteria for IPv6 Packets, page 5](#) (required)
- [Using the Match Criteria to Manage IPv6 Traffic Flows, page 6](#) (required)
- [Verifying Packet Marking Criteria, page 8](#) (optional)
- [Confirming the Service Policy, page 13](#) (optional)

Restrictions for Classifying Traffic in IPv6 Networks

Except for the modifications to the **match dscp** and **match precedence** commands and the addition of the IPv6-specific **match access-group name** command, the functionality of all of the **match** commands is the same for both IPv4 and IPv6.

The **set cos** and **match cos** commands for 802.1Q (dot1Q) interfaces are supported only for Cisco Express Forwarding-switched packets. Process-switched packets, such as router-generated packets, are not marked when these options are used.

Specifying Marking Criteria for IPv6 Packets

The following task establishes the match criteria (or marks the packets) that will be used later to match packets for classifying network traffic.

SUMMARY STEPS

1. **enable**
 2. **configure terminal**
 3. **policy map** *policy-map-name*
 4. **class** {*class-name* | **class-default**}
 5. **set precedence** {*precedence-value* | *from-field* [**table** *table-map-name*]}
- or
- set [ip] dscp** {*dscp-value* | *from-field* [**table** *table-map-name*]}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables such as privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	policy map <i>policy-map-name</i> Example: Router(config)# policy map policy1	Creates a policy map using the specified name and enters QoS policy-map configuration mode. <ul style="list-style-type: none">• Enter name of policy map you want to create.

	Command or Action	Purpose
Step 4	class { <i>class-name</i> class-default } Example: Router(config-pmap)# class class-default	Specifies the treatment for traffic of specified class (or the default class) and enters QoS policy-map class configuration mode.
Step 5	set precedence { <i>precedence-value</i> <i>from-field</i> [table <i>table-map-name</i>]} or set [ip] dscp { <i>dscp-value</i> <i>from-field</i> [table <i>table-map-name</i>]} Example: Router(config-pmap-c)# set dscp cos table table-map1 or Router(config-pmap-c)# set precedence cos table table-map1	Sets the precedence value. <ul style="list-style-type: none"> This example is based on the CoS value (and action) defined in the specified table map. Both precedence and DSCP cannot be changed in the same packets. Sets the DSCP value based on the CoS value (and action) defined in the specified table map.

Troubleshooting Tips

Confirm That Cisco Express Forwarding Is Enabled

Use the **show cef interface**, **show ipv6 cef**, **show ipv6 interface neighbors**, and **show interface statistics** commands to confirm that Cisco Express Forwarding is enabled and that packets are being Cisco Express Forwarding-switched.

Confirm That Packets Are Cisco Express Forwarding-Switched

Use the **show policy-map interface** command to display per-interface, per-policy Cisco Express Forwarding-switching statistics.

Using the Match Criteria to Manage IPv6 Traffic Flows

The following task describes how to use the **match** commands to match the traffic to the policies that you establish. You can use multiple match statements. Depending on the type of class, you can specify whether to match all classes or any of the classes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** {*class-name* | **class-default**}
4. **match precedence** *precedence-value* [*precedence-value precedence-value*]
or
match access-group name *ipv6-access-group*

or

```
match [ip] dscp dscp-value [dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value]
```

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables such as privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	class-map {class-name class-default } Example: Router(config-pmap-c)# class cls1	Creates the specified class and enters QoS class-map configuration mode.
Step 4	match precedence precedence-value [precedence-value precedence-value] or match access-group name ipv6-access-group or match [ip] dscp dscp-value [dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value dscp-value] Example: Router(config-pmap-c)# match precedence 5 or Router(config-pmap-c)# match access-group name ipv6acl or Router(config-pmap-c)# match ip dscp 15	Matches the precedence value. The precedence applies to both IPv4 and IPv6 packets. or Specifies the name of an IPv6 access list against whose contents packets are checked to determine if they belong to the traffic class. or Identifies a specific IP DSCP value as a match criterion.

Examples

The following example shows how to use the **match precedence** command to manage IPv6 traffic flows:

```
Router# configure terminal
```

Enter configuration commands, one per line. End with CNTL/Z.

```
Router(config)# class-m c1
```

```
Router(config-cmap)# match precedence 5
```

```

Router(config-cmap)# end
Router#
Router(config)# policy p1
Router(config-pmap)# class c1
Router(config-pmap-c)# police 10000 conform set-prec-trans 4

```

Verifying Packet Marking Criteria

To verify that packet marking is working as expected, use the `show policy` command. The interesting information from the output of this command is the difference in the number of total packets versus the number of packets marked.

```

Router# show policy p1

Policy Map p1
  Class c1
    police 10000 1500 1500 conform-action set-prec-transmit 4 exceed-action drop

Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# interface serial 4/1
Router(config-if)# service out p1
Router(config-if)# end

Router# show policy interface s4/1

Serial4/1
Service-policy output: p1
  Class-map: c1 (match-all)
    0 packets, 0 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: precedence 5
    police:
      10000 bps, 1500 limit, 1500 extended limit
      conformed 0 packets, 0 bytes; action: set-prec-transmit 4
      exceeded 0 packets, 0 bytes; action: drop
      conformed 0 bps, exceed 0 bps violate 0 bps

  Class-map: class-default (match-any)
    10 packets, 1486 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any

```

Interpreting Packet Counters in show policy-map interface Command Output

During periods of transmit congestion at the outgoing interface, packets arrive faster than the interface can send them. It is helpful to know how to interpret the output of the `show policy-map interface` command, which is useful for monitoring the results of a service-policy created with Cisco's modular QoS CLI.

Congestion typically occurs when a fast ingress interface feeds a relatively slow egress interface. A common congestion point is a branch-office router with an Ethernet port facing the LAN and a serial port facing the WAN. Users on the LAN segment are generating 10 Mbps of traffic, which is being fed into a T1 with 1.5 Mbps of bandwidth.

Functionally, congestion is defined as filling the transmit ring on the interface (a ring is a special buffer control structure). Every interface supports a pair of rings: a receive ring for receiving packets and a transmit ring for sending packets. The size of the rings varies with the interface controller and with the bandwidth of the interface or virtual circuit (VC). As in the following example, use the **show atm vc vcd** command to display the value of the transmit ring on a PA-A3 ATM port adapter.

```
Router# show atm vc 3

ATM5/0.2: VCD: 3, VPI: 2, VCI: 2
VBR-NRT, PeakRate: 30000, Average Rate: 20000, Burst Cells: 94
AAL5-LLC/SNAP, etype:0x0, Flags: 0x20, VCmode: 0x0
OAM frequency: 0 second(s)
PA TxRingLimit: 10
InARP frequency: 15 minutes(s)
Transmit priority 2
InPkts: 0, OutPkts: 0, InBytes: 0, OutBytes: 0
InProc: 0, OutProc: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
InPktDrops: 0, OutPktDrops: 0
CrcErrors: 0, SarTimeOuts: 0, OverSizedSDUs: 0
OAM cells received: 0
OAM cells sent: 0
Status: UP
```

Cisco IOS software (also referred to as the Layer 3 processor) and the interface driver use the transmit ring when moving packets to the physical media. The two processors collaborate in the following way:

- The interface sends packets according to the interface rate or a shaped rate.
- The interface maintains a hardware queue or transmit ring, where it stores the packets waiting for transmission onto the physical wire.
- When the hardware queue or transmit ring fills, the interface provides explicit back pressure to the Layer 3 processor system. It notifies the Layer 3 processor to stop dequeuing packets to the interface's transmit ring because the transmit ring is full. The Layer 3 processor now stores the excess packets in the Layer 3 queues.
- When the interface sends the packets on the transmit ring and empties the ring, it once again has sufficient buffers available to store the packets. It releases the back pressure, and the Layer 3 processor dequeues new packets to the interface.

The most important aspect of this communication system is that the interface recognizes that its transmit ring is full and throttles the receipt of new packets from the Layer 3 processor system. Thus, when the interface is congested, the drop decision is moved from a random, last-in, first-dropped decision in the first in, first out (FIFO) queue of the transmit ring to a differentiated decision based on IP-level service policies implemented by the Layer 3 processor.

Number of Packets and Packets Matched

Service policies apply only to packets stored in the Layer 3 queues. [Table 1](#) illustrates which packets sit in the Layer 3 queue. Locally generated packets are always process switched and are delivered first to the Layer 3 queue before being passed on to the interface driver. Fast-switched and Cisco Express Forwarding-switched packets are delivered directly to the transmit ring and sit in the L3 queue only when the transmit ring is full.

Table 1 **Packet Types and the Layer 3 Queue**

Packet Type	Congestion	Noncongestion
Locally generated packets, including Telnet packets and pings	Yes	Yes
Other packets that are process switched	Yes	Yes
Packets that are Cisco Express Forwarding- or fast-switched	Yes	No

The following example shows these guidelines applied to the **show policy-map interface** command output. The four key counters are shown in boldface type.

Router# **show policy-map interface atm 1/0.1**

```

ATM1/0.1: VC 0/100 -
Service-policy output: cbwfq (1283)
  Class-map: A (match-all) (1285/2)
    28621 packets, 7098008 bytes
    5 minute offered rate 10000 bps, drop rate 0 bps
    Match: access-group 101 (1289)
    Weighted Fair Queueing
      Output Queue: Conversation 73
      Bandwidth 500 (kbps) Max Threshold 64 (packets)
      (pkts matched/bytes matched) 28621/7098008
      (depth/total drops/no-buffer drops) 0/0/0
  Class-map: B (match-all) (1301/4)
    2058 packets, 148176 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: access-group 103 (1305)
    Weighted Fair Queueing
      Output Queue: Conversation 75
      Bandwidth 50 (kbps) Max Threshold 64 (packets)
      (pkts matched/bytes matched) 0/0
      (depth/total drops/no-buffer drops) 0/0/0
  Class-map: class-default (match-any) (1309/0)
    19 packets, 968 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any (1313)

```

[Table 2](#) defines the counters that appear in the example in boldfaced type.

Table 2 **Packet Counters from show policy-map interface Output**

Counter	Explanation
28621 packets, 7098008 bytes	The number of packets matching the criteria of the class. This counter increments whether or not the interface is congested.
(pkts matched/bytes matched) 28621/709800	The number of packets matching the criteria of the class when the interface was congested. In other words, the interface's transmit ring was full, and the driver and the L3 processor system worked together to queue the excess packets in the L3 queues, where the service policy applies. Packets that are process switched always go through the L3 queuing system and therefore increment the "packets matched" counter.
Class-map: B (match-all) (1301/4)	These numbers define an internal ID used with the CISCO-CLASS-BASED-QOS-MIB Management Information Base (MIB). They no longer appear in the show policy-map command output in current releases of Cisco IOS.
5 minute offered rate 0 bps, drop rate 0 bps	Use the load-interval command to change this value and make it a more instantaneous value. The lowest value is 30 seconds; however, statistics displayed in the show policy-map interface command output are updated every 10 seconds. Because the command effectively provides a snapshot at a specific moment, the statistics may not reflect a temporary change in queue size.

Without congestion, there is no need to queue any excess packets. When congestion occurs, packets, including Cisco Express Forwarding- and fast-switched packets, might go into the Layer 3 queue. If you use congestion management features, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled according to their assigned priority and the queueing mechanism configured for the interface.

Normally, the packets counter is much larger than the packets matched counter. If the values of the two counters are nearly equal, then the interface is receiving a large number of process-switched packets or is heavily congested. Both of these conditions should be investigated to ensure optimal packet forwarding.

Conversation Number Allocation

Routers allocate conversation numbers for the queues that are created when the service policy is applied. The following example shows the queues and related information.

```
Router# show policy-map interface s1/0.1 dlc1 100
```

```
Serial1/0.1: DLCI 100 -
output : mypolicy
Class voice
  Weighted Fair Queueing
  Strict Priority
  Output Queue: Conversation 72
```

```

    Bandwidth 16 (kbps) Packets Matched 0
    (pkts discards/bytes discards) 0/0
Class immediate-data
  Weighted Fair Queueing
    Output Queue: Conversation 73
      Bandwidth 60 (%) Packets Matched 0
      (pkts discards/bytes discards/tail drops) 0/0/0
      mean queue depth: 0
      drops: class random tail min-th max-th mark-prob
              0      0      0    64    128    1/10
              1      0      0    71    128    1/10
              2      0      0    78    128    1/10
              3      0      0    85    128    1/10
              4      0      0    92    128    1/10
              5      0      0    99    128    1/10
              6      0      0   106    128    1/10
              7      0      0   113    128    1/10
              rsvp    0      0   120    128    1/10
Class priority-data
  Weighted Fair Queueing
    Output Queue: Conversation 74
      Bandwidth 40 (%) Packets Matched 0 Max Threshold 64 (packets)
      (pkts discards/bytes discards/tail drops) 0/0/0
Class class-default
  Weighted Fair Queueing
    Flow Based Fair Queueing
      Maximum Number of Hashed Queues 64 Max Threshold 20 (packets)

```

Information reported for each class includes the following:

- Class definition
- Queueing method applied
- Output Queue Conversation number
- Bandwidth used
- Number of packets discarded
- Number of bytes discarded
- Number of packets dropped

The **class-default** class is the default class to which traffic is directed, if that traffic does not satisfy the match criteria of other classes whose policy is defined in the policy map. The **fair-queue** command allows you to specify the number of dynamic queues into which IP flows are sorted and classified. Alternately, routers allocate a default number of queues derived from the bandwidth on the interface or VC. Supported values in either case are a power of two, in a range from 16 to 4096.

[Table 3](#) lists the default values for interfaces and for ATM permanent virtual circuits (PVCs).

Table 3 *Default Number of Dynamic Queues as a Function of Interface Bandwidth*

Bandwidth Range	Number of Dynamic Queues
Less than or equal to 64 kbps	16
More than 64 kbps and less than or equal to 128 kbps	32
More than 128 kbps and less than or equal to 256 kbps	64

Table 3 *Default Number of Dynamic Queues as a Function of Interface Bandwidth*

Bandwidth Range	Number of Dynamic Queues
More than 256 kbps and less than or equal to 512 kbps	128
More than 512 kbps	256

[Table 4](#) lists the default number of dynamic queues in relation to ATM PVC bandwidth.

Table 4 *Default Number of Dynamic Queues as a Function of ATM PVC Bandwidth*

Bandwidth Range	Number of Dynamic Queues
Less than or equal to 128 kbps	16
More than 128 kbps and less than or equal to 512 kbps	32
More than 512 kbps and less than or equal to 2000 kbps	64
More than 2000 kbps and less than or equal to 8000 kbps	128
More than 8000 kbps	256

Based on the number of reserved queues for WFQ, Cisco IOS software assigns a conversation or queue number as shown in [Table 5](#).

Table 5 *Conversation Numbers Assigned to Queues*

Number	Type of Traffic
1 to 256	General flow-based traffic queues. Traffic that does not match to a user-created class will match to class-default and one of the flow-based queues.
257 to 263	Reserved for Cisco Discovery Protocol (formerly known as CDP) and for packets marked with an internal high-priority flag.
264	Reserved queue for the priority class (classes configured with the priority command). Look for the “Strict Priority” value for the class in the show policy-map interface output. The priority queue uses a conversation ID equal to the number of dynamic queues, plus 8.
265 and higher	Queues for user-created classes.

Confirming the Service Policy

This task tests the packets matched counter and your service policy. Ensure that the traffic flow matches the input or output parameter of the policy. For example, downloading a file from an FTP server generates congestion in the receive direction because the server sends large MTU-sized frames, and the client PC returns small acknowledgments (ACKs).

Before you begin this task, simulate congestion with an extended ping using a large ping size and a large number of pings. Also, try downloading a large file from an FTP server. The file constitutes “disturbing” data and fills the interface bandwidth.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface atm slot/0. subinterface-number {multipoint | point-to-point}**
4. **ip address ip-address mask [secondary]**
5. **pvc [name] vpi/vci [ces | ilmi | qsaal | smds]**
6. **tx-ring-limit ring-limit**
7. **service-policy {input | output} policy-map-name**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface atm slot/0. subinterface-number {multipoint point-to-point} Example: Router(config)# interface atm 1/0.1 point-to-point}	Enters interface configuration mode.
Step 4	ip address ip-address mask [secondary] Example: Router(config-if)# ip address 10.1.1.1 255.255.255.0	Specifies the IP address of the interface you want to test.
Step 5	pvc [name] vpi/vci [ces ilmi qsaal smds] Example: Router(config-if)# pvc cisco 0/5	Creates or assigns a name to an ATM PVC, optionally specifies the encapsulation type on an ATM PVC, and enters interface-ATM-VC configuration mode.

	Command or Action	Purpose
Step 6	tx-ring-limit <i>ring-limit</i> Example: Router(config-if-atm-vc)# tx-ring-limit 10	Reduces the size of the transmit ring of the interface. Lowering this value accelerates the use of the QoS in the Cisco IOS software. <ul style="list-style-type: none"> Specify the ring limit as the number of packets for 2600 and 3600 series routers, or as the number of memory particles for 7200 and 7500 series routers.
Step 7	service-policy { input output } <i>policy-map-name</i> Example: Router(config-if-atm-vc)# service-policy output policy9	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC. <ul style="list-style-type: none"> Note that the packets matched counter is a part of queueing feature and is available only on service policies attached in output direction.

Configuration Examples for Implementing QoS for IPv6

This section provides the following configuration examples:

- [Verification of Cisco Express Forwarding Switching: Example, page 15](#)
- [Matching DSCP Value: Example, page 16](#)

Verification of Cisco Express Forwarding Switching: Example

The following is sample output from the **show cef interface detail** command for Ethernet interface 1/0/0. Use this command to verify that Cisco Express Forwarding switching is enabled for policy decisions to occur. Notice that the display shows that Cisco Express Forwarding switching is enabled.

Router# **show cef interface Ethernet 1/0/0 detail**

```
Ethernet1/0/0 is up (if_number 9)
Corresponding hwidb fast_if_number 9
Corresponding hwidb firstsw->if_number 9
Internet address is 10.2.61.8/24
ICMP redirects are always sent
Per packet load-sharing is disabled
IP unicast RPF check is disabled
Inbound access list is not set
Outbound access list is not set
IP policy routing is disabled
Hardware idb is Ethernet1/0/0
Fast switching type 1, interface type 5
IP Distributed CEF switching enabled
IP Feature Fast switching turbo vector
IP Feature CEF switching turbo vector
Input fast flags 0x0, Output fast flags 0x0
ifindex 7(7)
Slot 1 Slot unit 0 VC -1
Transmit limit accumulator 0x48001A82 (0x48001A82)
IP MTU 1500
```

Matching DSCP Value: Example

The following example shows how to configure the service policy called `priority50` and attach service policy `priority50` to an interface. In this example, the **match dscp** command includes the optional **ip** keyword, meaning that the match is for IPv4 packets only. The class map called `ipdscp15` will evaluate all packets entering interface Fast Ethernet 1/0/0. If the packet is an IPv4 packet and has a DSCP value of 15, the packet will be treated as priority traffic and will be allocated with bandwidth of 50 kbps.

```
Router(config)# class-map ipdscp15
Router(config-cmap)# match ip dscp 15
Router(config)# exit
Router(config)# policy-map priority50
Router(config-pmap)# class ipdscp15
Router(config-pmap-c)# priority 50
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy input priority55
```

To match on IPv6 packets only, use the **match dscp** command without the **ip** keyword preceded by the **match protocol** command. Ensure that the class map has the **match-all** attribute (which is the default).

```
Router(config)# class-map ipdscp15
Router(config-cmap)# match protocol ipv6
Router(config-cmap)# match dscp 15
Router(config)# exit
```

To match packets on both IPv4 and IPv6 protocols, use the **match dscp** command:

```
Router(config)# class-map ipdscp15
Router(config-cmap)# match dscp 15
Router(config)# exit
```

Additional References

The following sections provide references related to the Implementing QoS for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2474	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>
RFC 2475	<i>An Architecture for Differentiated Services Framework</i>
RFC 2597	<i>Assured Forwarding PHB</i>
RFC 2598	<i>An Expedited Forwarding PHB</i>
RFC 2640	<i>Internet Protocol, Version 6 Specification</i>
RFC 2697	<i>A Single Rate Three Color Marker</i>
RFC 2698	<i>A Two Rate Three Color Marker</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **match access-group name**
- **match dscp**
- **match precedence**
- **set dscp**
- **set precedence**

Feature Information for Implementing QoS for IPv6

Table 6 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 6 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 6 Feature Information for Implementing QoS for IPv6

Feature Name	Releases	Feature Information
IPv6 quality of service (QoS)	12.0(28)S ¹ 12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	QoS features supported for IPv6 environments include packet classification, queueing, traffic shaping, WRED, class-based packet marking, and policing of IPv6 packets.
IPv6 QoS: MQC packet classification	12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>The modular QoS CLI allows you to define traffic classes, create and configure traffic policies (policy maps), and then attach those traffic policies to interfaces.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Packet Classification in IPv6, page 3 • Specifying Marking Criteria for IPv6 Packets, page 5

Table 6 *Feature Information for Implementing QoS for IPv6 (continued)*

Feature Name	Releases	Feature Information
IPv6 QoS: MQC traffic shaping	12.0(28)S ¹ 12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Traffic shaping allows you to limit the packet dequeue rate by holding additional packets in the queues and forwarding them as specified by parameters configured for traffic shaping features. The following sections provide information about this feature: <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Traffic Policing in IPv6 Environments, page 4 • Interpreting Packet Counters in show policy-map interface Command Output, page 8
IPv6 QoS: MQC traffic policing	12.0(28)S ¹ 12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Configuration or command usage for policing are the same in IPv6 environments as for IPv4 environments. The following sections provide information about this feature: <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Traffic Policing in IPv6 Environments, page 4
IPv6 QoS: MQC packet marking/re-marking	12.0(28)S ¹ 12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Class-based marking allows you to set the IPv6 precedence and DSCP values for traffic management. The following sections provide information about this feature: <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Policies and Class-Based Packet Marking in IPv6 Networks, page 3 • Traffic Policing in IPv6 Environments, page 4

Table 6 **Feature Information for Implementing QoS for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 QoS: queueing	12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Class-based and flow-based queueing are supported for IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Congestion Management in IPv6 Networks, page 3 • Traffic Policing in IPv6 Environments, page 4 • Interpreting Packet Counters in show policy-map interface Command Output, page 8
IPv6 QoS: MQC WRED-based drop	12.0(28)S ¹ 12.2(33)SRA 12.2(18)SXE ² 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	WRED implements the RED-based drop policy on the packets that are likely to overflow the limits of CBWFQ. The following sections provide information about this feature: <ul style="list-style-type: none"> • Implementation Strategy for QoS for IPv6, page 2 • Congestion Avoidance for IPv6 Traffic, page 4

1. Feature is supported on Cisco 12000 series Internet routers in Cisco IOS Release 12.0(28)S.
2. Cisco IOS Release 12.2(18)SXE provides support for this feature. Cisco IOS Release 12.2(18)SXE is specific to Cisco Catalyst 6500 and Cisco 7600 series routers.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2002–2009 Cisco Systems, Inc. All rights reserved.



Implementing RIP for IPv6

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure Routing Information Protocol for IPv6. RIP is a distance-vector routing protocol that uses hop count as a routing metric. RIP is an Interior Gateway Protocol (IGP) most commonly used in smaller networks.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing RIP for IPv6”](#) section on page 16.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing RIP for IPv6, page 2](#)
- [Information About Implementing RIP for IPv6, page 2](#)
- [How to Implement RIP for IPv6, page 3](#)
- [Configuration Examples for IPv6 RIP, page 13](#)
- [Additional References, page 13](#)
- [Command Reference, page 15](#)
- [Feature Information for Implementing RIP for IPv6, page 16](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2001–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing RIP for IPv6

- This module assumes that you are familiar with IPv6 addressing and basic configuration. Refer to [Implementing IPv6 Addressing and Basic Connectivity](#) for more information.
- This module assumes that you are familiar with IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for IPv4 configuration and command reference information, as needed.

Information About Implementing RIP for IPv6

To configure IPv6 RIP, you need to understand the following concepts:

- [RIP for IPv6, page 2](#)
- [Nonstop Forwarding for IPv6 RIP, page 2](#)

RIP for IPv6

IPv6 RIP functions the same and offers the same benefits as RIP in IPv4. RIP enhancements for IPv6, detailed in RFC 2080, include support for IPv6 addresses and prefixes, and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages. New commands specific to RIP in IPv6 were also added to the Cisco IOS command-line interface (CLI).

In the Cisco IOS software implementation of IPv6 RIP each IPv6 RIP process maintains a local routing table, referred to as a Routing Information Database (RIB). The IPv6 RIP RIB contains a set of best-cost IPv6 RIP routes learned from all its neighboring networking devices. If IPv6 RIP learns the same route from two different neighbors, but with different costs, it will store only the lowest cost route in the local RIB. The RIB also stores any expired routes that the RIP process is advertising to its neighbors running RIP. IPv6 RIP will try to insert every non-expired route from its local RIB into the master IPv6 RIB. If the same route has been learned from a different routing protocol with a better administrative distance than IPv6 RIP, the RIP route will not be added to the IPv6 RIB but the RIP route will still exist in the IPv6 RIP RIB.

Nonstop Forwarding for IPv6 RIP

Cisco nonstop forwarding (NSF) continues forwarding packets while routing protocols converge, therefore avoiding a route flap on switchover. When an RP failover occurs, the Forwarding Information Base (FIB) marks installed paths as stale by setting a new epoch. Subsequently, the routing protocols reconverge and populate the RIB and FIB. Once all NSF routing protocols converge, any stale routes held in the FIB are removed. A failsafe timer is required to delete stale routes, in case of routing protocol failure to repopulate the RIB and FIB.

RIP registers as an IPv6 NSF client. Doing so has the benefit of using RIP routes installed in the Cisco Express Forwarding table until RIP has converged on the standby.

How to Implement RIP for IPv6

When configuring supported routing protocols in IPv6, you must create the routing process, enable the routing process on interfaces, and customize the routing protocol for your particular network.

**Note**

The following sections describe the configuration tasks for creating an IPv6 RIP routing process and enabling the routing process on interfaces. The following sections do not provide in-depth information on customizing RIP because the protocol functions the same in IPv6 as it does in IPv4. Refer to the publications referenced in the “[Related Documents](#)” section for further IPv6 and IPv4 configuration and command reference information.

The tasks in the following sections explain how to configure IPv6 RIP. Each task in the list is identified as either required or optional:

This section contains the following procedures:

- [Enabling IPv6 RIP, page 3](#) (required)
- [Customizing IPv6 RIP, page 4](#) (optional)
- [Redistributing Routes into an IPv6 RIP Routing Process, page 6](#) (optional)
- [Configuring Route Tags for IPv6 RIP Routes, page 7](#) (optional)
- [Filtering IPv6 RIP Routing Updates, page 8](#) (optional)
- [Verifying IPv6 RIP Configuration and Operation, page 10](#) (optional)

Enabling IPv6 RIP

This task explains how to enable the specified IPv6 RIP process on an interface.

Prerequisites

Before configuring the router to run IPv6 RIP, globally enable IPv6 using the **ipv6 unicast-routing** command in global configuration mode, and enable IPv6 on any interfaces on which IPv6 RIP is to be enabled. For details on basic IPv6 connectivity tasks, refer to the *Implementing Basic Connectivity for IPv6* module.

If you want to set or change a global value, follow steps 1 and 2, and then use the optional **ipv6 router rip** command in global configuration mode (see [Customizing IPv6 RIP, page 4](#) for an example).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 unicast-routing**
4. **interface** *type number*
5. **ipv6 rip** *name* **enable**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 4	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 5	ipv6 rip <i>name</i> enable Example: Router(config-if)# ipv6 rip process1 enable	Enables the specified IPv6 RIP routing process on an interface.

Customizing IPv6 RIP

This optional task explains how to customize IPv6 RIP by configuring the maximum numbers of equal-cost paths that IPv6 RIP will support, adjusting the IPv6 RIP timers, and originating a default IPv6 route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 router rip** *word*
4. **maximum-paths** *number-paths*
5. **exit**
6. **interface** *type number*
7. **ipv6 rip** *name* **default-information** { **only** | **originate** } [**metric** *metric-value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 router rip word Example: Router(config)# ipv6 router rip process1	Configures an IPv6 RIP routing process and enters router configuration mode for the IPv6 RIP routing process. <ul style="list-style-type: none"> Use the <i>word</i> argument to identify a specific IPv6 RIP routing process.
Step 4	maximum-paths number-paths Example: Router(config-router)# maximum-paths 1	(Optional) Defines the maximum number of equal-cost routes that IPv6 RIP can support. <ul style="list-style-type: none"> The <i>number-paths</i> argument is an integer from 1 to 64. The default for RIP is four paths.
Step 5	exit Example: Router(config-if)# exit	Exits interface configuration mode and enters global configuration mode.
Step 6	interface type number Example: Router(config)# interface Ethernet 0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 7	ipv6 rip name default-information {only originate} [metric metric-value] Example: Router(config-if)# ipv6 rip process1 default-information originate	(Optional) Originates the IPv6 default route (::/0) into the specified RIP routing process updates sent out of the specified interface. <p>Note To avoid routing loops after the IPv6 default route (::/0) is originated out of any interface, the routing process ignores all default routes received on any interface.</p> <ul style="list-style-type: none"> Specifying the only keyword originates the default route (::/0) but suppresses all other routes in the updates sent on this interface. Specifying the originate keyword originates the default route (::/0) in addition to all other routes in the updates sent on this interface.

Redistributing Routes into an IPv6 RIP Routing Process

RIP supports the use of a route map to select routes for redistribution. Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map “match tag” function.

The maximum metric that RIP can advertise is 16, and a metric of 16 denotes a route that is unreachable. Therefore, if you are redistributing routes with metrics greater than or equal to 16, then by default RIP will advertise them as unreachable. These routes will not be used by neighboring routers. The user must configure a redistribution metric of less than 15 for these routes.



Note

You must to advertise a route with metric of 15 or less. A RIP router always adds an interface cost—the default is 1—onto the metric of a received route. If you advertise a route with metric 15, your neighbor will add 1 to it, making a metric of 16. Because a metric of 16 is unreachable, your neighbor will not install the route in the routing table.

If no metric is specified, then the current metric of the route is used. To find the current metric of the route, enter the **show ipv6 route** command.

This task explains how to redistribute routes into an IPv6 RIP routing process.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 rip** *name* **enable**
5. **redistribute** *protocol* [*process-id*] {**level-1** | **level-1-2** | **level-2**} [**metric** *metric-value*] [**metric-type** {**internal** | **external**}] [**route-map** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface Ethernet 0/0	Specifies the interface type and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 4	<code>ipv6 rip word enable</code> Example: Router(config-if)# ipv6 router one enable	Enables an IPv6 Routing Information Protocol (RIP) routing process on an interface.
Step 5	<code>redistribute protocol [process-id] {level-1 level-1-2 level-2} [metric metric-value] [metric-type {internal external}] [route-map map-name]</code> Example: Router(config-router)# redistribute bgp 65001 route-map bgp-to-rip	Redistributes the specified routes into the IPv6 RIP routing process. <ul style="list-style-type: none"> The <i>protocol</i> argument can be one of the following keywords: bgp, connected, isis, rip, or static. The rip keyword and <i>process-id</i> argument specify an IPv6 RIP routing process. <p>Note The connected keyword refers to routes that are established automatically by assigning IPv6 addresses to an interface.</p>

Configuring Route Tags for IPv6 RIP Routes

When performing route redistribution, you can associate a numeric tag with a route. The tag is advertised with the route by RIP and will be installed along with the route in neighboring router's routing table.

If you redistribute a tagged route (for example, a route in the IPv6 routing table that already has a tag) into RIP, then RIP will automatically advertise the tag with the route. If you use a redistribution route map to specify a tag, then RIP will use the route map tag in preference to the routing table tag.

The following task explains how to set route tags using a route map.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **route-map map-tag [permit | deny] [sequence-number]**
4. **match ipv6 address {prefix-list prefix-list-name | access-list-name}**
5. **set tag tag-value**

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code> Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	<code>configure terminal</code> Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	route-map <i>map-tag</i> [permit deny] <i>[sequence-number]</i> Example: Router(config)# route-map bgp-to-rip permit 10	Defines a route map, and enters route-map configuration mode. <ul style="list-style-type: none"> Follow this step with a match command.
Step 4	match ipv6 address (prefix-list <i>prefix-list-name</i> <i>access-list-name</i>) Example: Router(config-route-map)# match ipv6 address prefix-list bgp-to-rip-flt	Specifies a list of IPv6 prefixes to be matched.
Step 5	set tag <i>tag-value</i> Example: Router(config-route-map)# set tag 4	Sets the tag value to associate with the redistributed routes.

Filtering IPv6 RIP Routing Updates

Route filtering using distribute lists provides control over the routes RIP receives and advertises. This control may be exercised globally or per interface.

This task explains how to filter IPv6 RIP routing updates by applying a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

IPv6 Distribute Lists

Filtering is controlled by distribute lists. Input distribute lists control route reception, and input filtering is applied to advertisements received from neighbors. Only those routes that pass input filtering will be inserted in the RIP local routing table and become candidates for insertion into the IPv6 routing table.

Output distribute lists control route advertisement; Output filtering is applied to route advertisements sent to neighbors. Only those routes passing output filtering will be advertised.

Global distribute lists (which are distribute lists that do not apply to a specified interface) apply to all interfaces. If a distribute list specifies an interface, then that distribute list applies only to that interface.

An interface distribute list always takes precedence. For example, for a route received at an interface, with the interface filter set to deny, and the global filter set to permit, the route is blocked, the interface filter is passed, the global filter is blocked, and the route is passed.

IPv6 Prefix List Operand Keywords

IPv6 prefix lists are used to specify certain prefixes or a range of prefixes that must be matched before a permit or deny statement can be applied. Two operand keywords can be used to designate a range of prefix lengths to be matched. A prefix length of less than, or equal to, a value is configured with the **le** keyword. A prefix length greater than, or equal to, a value is specified using the **ge** keyword. The **ge** and **le** keywords can be used to specify the range of the prefix length to be matched in more detail than the usual *ipv6-prefix/prefix-length* argument. For a candidate prefix to match against a prefix list entry three conditions can exist:

- The candidate prefix must match the specified prefix list and prefix length entry.

- The value of the optional **le** keyword specifies the range of allowed prefix lengths from the *prefix-length* argument up to, and including, the value of the **le** keyword.
- The value of the optional **ge** keyword specifies the range of allowed prefix lengths from the value of the **ge** keyword up to, and including, 128.

**Note**

Note that the first condition must match before the other conditions take effect.

An exact match is assumed when the **ge** or **le** keywords are not specified. If only one keyword operand is specified then the condition for that keyword is applied, and the other condition is not applied. The *prefix-length* value must be less than the **ge** value. The **ge** value must be less than, or equal to, the **le** value. The **le** value must be less than or equal to 128.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**deny** *ipv6-prefix/prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*]
4. **ipv6 prefix list** *prefix-list-name* [**seq** *seq-number*] {**permit** *ipv6-prefix/prefix-length* | **description** *text*} [**ge** *ge-value*] [**le** *le-value*]
5. Repeat Steps 3 and 4 as many times as necessary to build the prefix list.
6. **ipv6 router rip** *name*
7. **distribute-list prefix-list** *prefix-list-name* {**in** | **out**} [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	
Step 3	ipv6 prefix list <i>prefix-list-name</i> [seq <i>seq-number</i>] { deny <i>ipv6-prefix/prefix-length</i> description <i>text</i> } [ge <i>ge-value</i>] [le <i>le-value</i>]	Creates an entry in the IPv6 prefix list.
	Example: Router(config)# ipv6 prefix-list abc permit 2001:0db8::/16	

	Command or Action	Purpose
Step 4	ipv6 prefix list <i>prefix-list-name</i> [seq <i>seq-number</i>] [deny <i>ipv6-prefix/prefix-length</i> description <i>text</i>] [ge <i>ge-value</i>] [le <i>le-value</i>] Example: Router(config)# ipv6 prefix-list abc deny ::/0	Creates an entry in the IPv6 prefix list.
Step 5	Repeat Steps 3 and 4 as many times as necessary to build the prefix list.	—
Step 6	ipv6 router rip <i>name</i> Example: Router(config)# ipv6 router rip process1	Configures an IPv6 RIP routing process.
Step 7	distribute-list prefix-list <i>prefix-list-name</i> [in out] [<i>interface-type interface-number</i>] Example: Router(config-rtr-rip)# distribute-list prefix-list process1 in ethernet 0/0	Applies a prefix list to IPv6 RIP routing updates that are received or sent on an interface.

Verifying IPv6 RIP Configuration and Operation

A user may want to check IPv6 RIP configuration and operation. This task explains how to display information to verify the configuration and operation of IPv6 RIP.

SUMMARY STEPS

1. **show ipv6 rip** [*name*] [**database** | **next-hops**]
2. **show ipv6 route** [*ipv6-address* | *ipv6-prefix/prefix-length* | *protocol* | *interface-type interface-number*]
3. **enable**
4. **debug ipv6 rip** [*interface-type interface-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	show ipv6 rip [<i>name</i>] [database next-hops] Example: Router> show ipv6 rip process1 database	(Optional) Displays information about current IPv6 RIP processes. <ul style="list-style-type: none"> In this example, IPv6 RIP process database information is displayed for the specified IPv6 RIP process.
Step 2	show ipv6 route [<i>ipv6-address</i> <i>ipv6-prefix/prefix-length</i> <i>protocol</i> <i>interface-type interface-number</i>] Example: Router> show ipv6 route rip	(Optional) Displays the current contents of the IPv6 routing table. <ul style="list-style-type: none"> In this example, only IPv6 RIP routes are displayed.

	Command or Action	Purpose
Step 3	enable Example: Router> enable	Enables higher privilege levels, such as privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 4	debug ipv6 rip [<i>interface-type</i> <i>interface-number</i>] Example: Router# debug ipv6 rip	(Optional) Displays debugging messages for IPv6 RIP routing transactions.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 rip Command, page 11](#)
- [Sample Output from the show ipv6 route Command, page 12](#)
- [Sample Output from the debug ipv6 rip Command, page 12](#)

Sample Output from the show ipv6 rip Command

In the following example, output information about all current IPv6 RIP processes is displayed using the **show ipv6 rip** command:

```
Router> show ipv6 rip

RIP process "process1", port 521, multicast-group FF02::9, pid 62
  Administrative distance is 120. Maximum paths is 1
  Updates every 5 seconds, expire after 15
  Holddown lasts 10 seconds, garbage collect after 30
  Split horizon is on; poison reverse is off
  Default routes are generated
  Periodic updates 223, trigger updates 1
Interfaces:
  Ethernet0/0
Redistribution:
  Redistributing protocol bgp 65001 route-map bgp-to-rip
```

In the following example, output information about a specified IPv6 RIP process database is displayed using the **show ipv6 rip** command with the *name* argument and the **database** keyword. In the following output for the IPv6 RIP process named process1, timer information is displayed, and route 2001:0db8::16/64 has a route tag set:

```
Router> show ipv6 rip process1 database

RIP process "process1", local RIB
2001:0db8::/64, metric 2
  Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
2001:0db8::/16, metric 2 tag 4, installed
  Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
2001:0db8:1::/16, metric 2 tag 4, installed
  Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
2001:0db8:2::/16, metric 2 tag 4, installed
  Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
::/0, metric 2, installed
  Ethernet0/0/FE80::A8BB:CCFF:FE00:B00, expires in 13 secs
```

In the following example, output information for a specified IPv6 RIP process is displayed using the **show ipv6 rip** user EXEC command with the *name* argument and the **next-hops** keyword:

```
Router> show ipv6 rip process1 next-hops

RIP process "process1", Next Hops
  FE80::A8BB:CCFF:FE00:A00/Ethernet0/0 [4 paths]
```

Sample Output from the show ipv6 route Command

The current metric of the route can be found by entering the **show ipv6 route** command. In the following example, output information for all IPv6 RIP routes is displayed using the **show ipv6 route** command with the **rip** protocol keyword:

```
Router> show ipv6 route rip

IPv6 Routing Table - 17 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
R   2001:0db8:1::/32 [120/2]
    via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
R   2001:0db8:2::/32 [120/2]
    via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
R   2001:0db8:3::/32 [120/2]
    via FE80::A8BB:CCFF:FE00:A00, Ethernet0/0
```

Sample Output from the debug ipv6 rip Command

In the following example, debugging messages for IPv6 RIP routing transactions are displayed using the **debug ipv6 rip** command:



Note

By default, the system sends the output from **debug** commands and system error messages to the console. To redirect debugging output, use the **logging** command options within privileged EXEC mode. Possible destinations include the console, virtual terminals, internal buffer, and UNIX hosts running a syslog server.

```
Router# debug ipv6 rip

RIPNg: Sending multicast update on Ethernet0/0 for process1
      src=FE80::A8BB:CCFF:FE00:B00
      dst=FF02::9 (Ethernet0/0)
      sport=521, dport=521, length=112
      command=2, version=1, mbz=0, #rte=5
      tag=0, metric=1, prefix=2001:0db8::/64
      tag=4, metric=1, prefix=2001:0db8:1::/16
      tag=4, metric=1, prefix=2001:0db8:2::/16
      tag=4, metric=1, prefix=2001:0db8:3::/16
      tag=0, metric=1, prefix=::/0
RIPNg: Next RIB walk in 10032
RIPNg: response received from FE80::A8BB:CCFF:FE00:A00 on Ethernet0/0 for process1
      src=FE80::A8BB:CCFF:FE00:A00 (Ethernet0/0)
      dst=FF02::9
      sport=521, dport=521, length=92
      command=2, version=1, mbz=0, #rte=4
      tag=0, metric=1, prefix=2001:0db8::/64
      tag=0, metric=1, prefix=2001:0db8:1::/32
      tag=0, metric=1, prefix=2001:0db8:2::/32
      tag=0, metric=1, prefix=2001:0db8:3::/32
```

Configuration Examples for IPv6 RIP

This section provides the following configuration example:

- [IPv6 RIP Configuration: Example, page 13](#)

IPv6 RIP Configuration: Example

In the following example, the IPv6 RIP process named process1 is enabled on the router and on Ethernet interface 0/0. The IPv6 default route (::/0) is advertised in addition to all other routes in router updates sent on Ethernet interface 0/0. Additionally, BGP routes are redistributed into the RIP process named process1 according to a route map where routes that match a prefix list are also tagged. The number of parallel paths is set to one to allow the route tagging, and the IPv6 RIP timers are adjusted. A prefix list named eth0/0-in-flt filters inbound routing updates on Ethernet interface 0/0.

```
ipv6 router rip process1
 maximum-paths 1
 redistribute bgp 65001 route-map bgp-to-rip
 distribute-list prefix-list eth0/0-in-flt in Ethernet0/0
!
interface Ethernet0/0
 ipv6 address 2001:0db8::/64 eui-64
 ipv6 rip process1 enable
 ipv6 rip process1 default-information originate
!
ipv6 prefix-list bgp-to-rip-flt seq 10 deny 2001:0db8:3::/16 le 128
ipv6 prefix-list bgp-to-rip-flt seq 20 permit 2001:0db8:1::/8 le 128
!
ipv6 prefix-list eth0/0-in-flt seq 10 deny ::/0
ipv6 prefix-list eth0/0-in-flt seq 15 permit ::/0 le 128
!
route-map bgp-to-rip permit 10
 match ipv6 address prefix-list bgp-to-rip-flt
 set tag 4
```

Where to Go Next

If you want to implement more IPv6 routing protocols, see the [Implementing IS-IS for IPv6](#) or [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing RIP for IPv6 feature.

Related Documents

Related Topic	Document Title
IPv4 RIP configuration tasks	“Configuring Routing Information Protocol,” <i>Cisco IOS IP Routing Protocols Configuration Guide</i>
RIP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	“RIP Commands,” <i>Cisco IOS IP Routing Protocols Command Reference</i>
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” <i>Cisco IOS IPv6 Configuration Guide</i>
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2080	<i>RIPng for IPv6</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<p>http://www.cisco.com/techsupport</p>

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **clear ipv6 rip**
- **debug ipv6 rip**
- **distribute-list prefix-list (IPv6 RIP)**
- **ipv6 rip default-information**
- **ipv6 rip enable**
- **ipv6 rip metric-offset**
- **ipv6 rip summary-address**
- **ipv6 router rip**
- **match ipv6 next-hop**
- **match ipv6 route-source**
- **poison-reverse (IPv6 RIP)**
- **port (IPv6 RIP)**
- **show ipv6 rip**
- **split-horizon (IPv6 RIP)**
- **timers (IPv6 RIP)**

Feature Information for Implementing RIP for IPv6

Table 10 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 10 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 10 Feature Information for Implementing RIP for IPv6

Feature Name	Releases	Feature Information
IPv6 routing: RIP for IPv6 (RIPng)	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	RIP enhancements for IPv6 include support for IPv6 addresses and prefixes, and the use of the all-RIP-routers multicast group address FF02::9 as the destination address for RIP update messages. This entire document provides information about this features.
IPv6 routing: route redistribution	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	Routes may be specified by prefix, using a route-map prefix list, or by tag, using the route-map “match tag” function. The following sections provide information about this feature: <ul style="list-style-type: none"> • Redistributing Routes into an IPv6 RIP Routing Process, page 6 • Configuring Route Tags for IPv6 RIP Routes, page 7 • IPv6 RIP Configuration: Example, page 13
IPv6 routing: RIPng nonstop forwarding	Cisco IOS XE Release 2.1	IPv6 RIP supports NSF. The following section provides information about this feature: <ul style="list-style-type: none"> • Nonstop Forwarding for IPv6 RIP, page 2

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing Traffic Filters and Firewalls for IPv6 Security

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure Cisco IOS IPv6 traffic filter and firewall features for your Cisco networking devices. These security features can protect your network from degradation or failure and also from data loss or compromised security resulting from intentional attacks and from unintended but damaging mistakes by well-meaning network users.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security” section on page 32](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [Information About Implementing Traffic Filters and Firewalls for IPv6 Security, page 2](#)
- [How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5](#)
- [Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26](#)
- [Additional References, page 29](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2001–2009 Cisco Systems, Inc. All rights reserved.

- [Command Reference, page 31](#)
- [Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security, page 32](#)

Prerequisites for Implementing Traffic Filters and Firewalls for IPv6 Security

You should be familiar with IPv6 addressing and basic configuration. Refer to the [Implementing IPv6 Addressing and Basic Connectivity](#) module for more information.

Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security

Cisco IOS Release 12.2(2)T through Cisco IOS Release 12.2(13)T and Cisco IOS Release 12.0(22)S and later releases support only standard IPv6 access control list (ACL) functionality. In Cisco IOS Release 12.0(23)S and 12.2(13)T or later releases, the standard IPv6 ACL functionality is extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

Information About Implementing Traffic Filters and Firewalls for IPv6 Security

To implement security features for IPv6, you need to understand the following concepts:

- [Access Control Lists for IPv6 Traffic Filtering, page 2](#)
- [Cisco IOS Firewall for IPv6, page 3](#)

Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode.

In Cisco IOS Release 12.0(23)S and 12.2(13)T or later releases, the standard IPv6 ACL functionality is extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

IPv6 ACL Extensions for IPsec Authentication Header

This feature provides the ability to match on the upper layer protocol (ULP) (for example, TCP, User Datagram Protocol [UDP], ICMP, SCTP) regardless of whether an authentication header (AH) is present or absent.

TCP or UDP traffic can be matched to the upper-layer protocol (ULP) (for example, TCP, UDP, ICMP, SCTP) if an AH is present or absent. Before this feature was introduced, this function was only available if an AH was absent.

This feature introduces the keyword **auth** to the **permit** and **deny** commands. The **auth** keyword allows matching traffic against the presence of the authentication header in combination with the specified protocol; that is, TCP or UDP.

IPv6 traffic can be matched to a ULP when an AH header is present. To perform this function, enter the **ahp** option for the *protocol* argument when using the **permit** or **deny** command.

Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the router based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local router address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local router address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

Cisco IOS Firewall for IPv6

The Cisco IOS Firewall feature provides advanced traffic filtering functionality as an integral part of a network's firewall. Cisco IOS Firewall for IPv6 enables you to implement Cisco IOS Firewall in IPv6 networks. Cisco IOS Firewall coexists with Cisco IOS Firewall for IPv4 networks and is supported on all dual-stack routers.

Cisco IOS Firewall for IPv6 features are as follows:

- Fragmented packet inspection—The fragment header is used to trigger fragment processing. Cisco IOS Firewall virtual fragment reassembly (VFR) examines out-of-sequence fragments and switches the packets into correct order, examines the number of fragments from a single IP given a unique identifier (Denial of Service [DoS] attack), and performs virtual reassembly to move packets to upper-layer protocols.
- IPv6 DoS attack mitigation—Mitigation mechanisms have been implemented in the same fashion as for IPv4 implementation, including SYN half-open connections.
- Tunneled packet inspection—Tunneled IPv6 packets terminated at a Cisco IOS firewall router can be inspected by the Cisco IOS Firewall for IPv6.
- Stateful packet inspection—The feature provides stateful packet inspection of TCP, UDP, Internet Control Message Protocol version 6 (ICMPv6), and FTP sessions.
- Stateful inspection of packets originating from the IPv4 network and terminating in an IPv6 environment—This feature uses IPv4-to-IPv6 translation services.
- Interpretation or recognition of most IPv6 extension header information—The feature provides IPv6 extension header information including routing header, hop-by-hop options header, and fragment header is interpreted or recognized.
- Port-to-application mapping (PAM)—Cisco IOS Firewall for IPv6 includes PAM.

PAM in Cisco IOS Firewall for IPv6

PAM allows you to customize TCP or UDP port numbers for network services or applications. PAM uses this information to support network environments that run services using ports that are different from the registered or well-known ports associated with an application.

Using the port information, PAM establishes a table of default port-to-application mapping information at the firewall. The information in the PAM table enables Context-based Access Control (CBAC) supported services to run on nonstandard ports. CBAC is limited to inspecting traffic using only the well-known or registered ports associated with an application, whereas PAM allows network administrators to customize network access control for specific applications and services.

PAM also supports host- or subnet-specific port mapping, which allows you to apply PAM to a single host or subnet using standard ACLs. Host- or subnet-specific port mapping is done using standard ACLs.

Cisco IOS Firewall Alerts, Audit Trails, and System Logging

Cisco IOS Firewall generates real-time alerts and audit trails based on events tracked by the firewall. Enhanced audit trail features use system logging to track all network transactions; to record time stamps, source host, destination host, and ports used; and to record the total number of transmitted bytes for advanced, session-based reporting. Real-time alerts send system logging error messages to central management consoles when the system detects suspicious activity. Using Cisco IOS Firewall inspection rules, you can configure alerts and audit trail information on a per-application protocol basis. For example, if you want to generate audit trail information for TCP traffic, you can specify the generation of this information in the Cisco IOS Firewall rule that defines TCP inspection.

The Cisco IOS Firewall provides audit trail messages to record details about inspected sessions. Audit trail information is configurable on a per-application basis using the CBAC inspection rules. To determine which protocol was inspected, use the port number associated with the responder. The port number appears immediately after the address.

IPv6 Packet Inspection

The following header fields are all used for IPv6 inspection—traffic class, flow label, payload length, next header, hop limit, and source or destination address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

Tunneling Support

IPv6 packets tunneled in IPv4 are not inspected. If a tunnel terminates on a router, and IPv6 traffic exiting the tunnel is nonterminating, then the traffic is inspected.

Virtual Fragment Reassembly

When VFR is enabled, VFR processing begins after ACL input lists are checked against incoming packets. The incoming packets are tagged with the appropriate VFR information.

Cisco IOS Firewall Restrictions

Cisco IOS Intrusion Detection System (IDS) is not supported for IPv6.

How to Implement Traffic Filters and Firewalls for IPv6 Security

The tasks in the following sections explain how to configure security features for IPv6:

- [Configuring IPv6 Traffic Filtering, page 5](#)
- [Controlling Access to a vty, page 8](#)
- [Configuring TCP or UDP Matching, page 11](#)
- [Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 12](#)
- [Configuring the Cisco IOS Firewall for IPv6, page 14](#)
- [Configuring the Cisco IOS Firewall for IPv6, page 14](#)
- [Verifying IPv6 Security Configuration and Operation, page 19](#)
- [Troubleshooting IPv6 Security Configuration and Operation, page 21](#)

Configuring IPv6 Traffic Filtering

The following sections describe how enable IPv6 traffic filtering:

- [Creating and Configuring an IPv6 ACL for Traffic Filtering, page 5](#)
- [Applying the IPv6 ACL to an Interface, page 7](#)

Restrictions

- If you are running Cisco IOS Release 12.2(13)T, 12.0(23)S, or later releases, proceed to the [“Creating and Configuring an IPv6 ACL for Traffic Filtering”](#) section. If you are running Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases, proceed to the [“Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases”](#) section.
- IPv6 ACLs are defined by a unique name (IPv6 does not support numbered ACLs). An IPv4 ACL and an IPv6 ACL cannot share the same name.

Creating and Configuring an IPv6 ACL for Traffic Filtering

This part describes how to configure your networking devices to filter traffic, function as a firewall, or detect potential viruses. The following task explains how to create an IPv6 ACL and configure the IPv6 ACL to filter traffic in Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases.

Prerequisites

In Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases, for backward compatibility, the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode is still supported; however, an IPv6 ACL defined with deny and permit conditions in global configuration mode is translated to IPv6 access list configuration mode. See the [“Creating and Applying IPv6 ACLs: Examples”](#) section for an example of a translated IPv6 ACL configuration.

Restrictions

- Each IPv6 ACL contains implicit permit rules to enable IPv6 neighbor discovery. These rules can be overridden by the user by placing a `deny ipv6` any any statement within an ACL. The IPv6 neighbor discovery process makes use of the IPv6 network layer service; therefore, by default, IPv6 ACLs implicitly allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, makes use of a separate data link layer protocol; therefore, by default, IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.
- Time-based and reflexive ACLs are not supported for IPv4 or IPv6 on the Cisco 12000 series platform. The **reflect**, **timeout**, and **time-range** keywords of the **permit** command in IPv6 are excluded on the Cisco 12000 series.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 access-list** *access-list-name*
- permit** *protocol* { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name*] [**timeout** *value*] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
or
deny *protocol* { *source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth** } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth** } [*operator* [*port-number*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	ipv6 access-list <i>access-list-name</i> Example: Router(config)# ipv6 access-list outbound	Defines an IPv6 ACL, and enters IPv6 access list configuration mode. <ul style="list-style-type: none"> The <i>access-list name</i> argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.
Step 4	permit protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth } [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth } [operator [port-number]] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number] [sequence value] [time-range name] or deny protocol {source-ipv6-prefix/prefix-length any host source-ipv6-address auth } [operator [port-number]] {destination-ipv6-prefix/prefix-length any host destination-ipv6-address auth } [operator [port-number]] [dest-option-type [doh-number doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number mh-type]] [routing] [routing-type routing-number] [sequence value] [time-range name] [undetermined-transport] Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 eq telnet any reflect reflectout or Example: Router(config-ipv6-acl)# deny tcp host 2001:0db8:1::1 any log-input	Specifies permit or deny conditions for an IPv6 ACL.

Applying the IPv6 ACL to an Interface

This task describes how to apply the IPv6 ACL to an interface in Cisco IOS Release 12.2(13)T and 12.0(23)S or later releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*

4. `ipv6 traffic-filter access-list-name {in | out}`

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface type number Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 traffic-filter access-list-name {in out} Example: Router(config-if)# ipv6 traffic-filter outbound out	Applies the specified IPv6 access list to the interface specified in the previous step.

Controlling Access to a vty

The following tasks explain how to restrict access to a vty on a router:

- [Creating an IPv6 ACL to Provide Access Class Filtering, page 8](#)
- [Applying an IPv6 ACL to the Virtual Terminal Line, page 10](#)

Creating an IPv6 ACL to Provide Access Class Filtering

The following task explains how to control access to a vty on a router by creating an IPv6 ACL to provide access class filtering.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list access-list-name**
4. **permit protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number] [sequence value] [time-range name]**

or

```
deny protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator
[port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth}
[operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label
value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [routing]
[routing-type routing-number] [sequence value] [time-range name] [undetermined-transport]
```

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	ipv6 access-list <i>access-list-name</i> Example: Router(config)# ipv6 access-list cisco	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.
Step 4	permit protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [reflect <i>name</i>] [timeout <i>value</i>] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] or deny protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] [undetermined-transport] Example: Router(config-ipv6-acl)# permit ipv6 host 2001:0DB8:0:4::32 any eq telnet or Example: Router(config-ipv6-acl)# deny ipv6 host 2001:0DB8:0:6::6/32 any	Specifies permit or deny conditions for an IPv6 ACL.

Applying an IPv6 ACL to the Virtual Terminal Line

After you have created the IPv6 ACL for access class filtering, you must apply it to a specified virtual terminal line. The following task describes how to apply the ACL to the virtual terminal line.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [*aux* | *console* | *tty* | *vty*] *line-number* [*ending-line-number*]
4. **ipv6 access-class** *ipv6-access-list-name* {*in* | *out*}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	line [aux console tty vty] <i>line-number</i> [<i>ending-line-number</i>] Example: Router(config)# line vty 0 4	Identifies a specific line for configuration and enters line configuration mode. <ul style="list-style-type: none"> In this example, the vty keyword is used to specify the virtual terminal lines for remote console access.
Step 4	ipv6 access-class <i>ipv6-access-list-name</i> { in out } Example: Router(config-line)# ipv6 access-class cisco in	Filters incoming and outgoing connections to and from the router based on an IPv6 ACL.

Configuring TCP or UDP Matching

TCP or UDP traffic can be matched to the ULP (for example, TCP, UDP, ICMP, SCTP) if an AH is present or absent. Before this feature was introduced, this function was only available if an AH was absent.

Use of the keyword **auth** with the **permit icmp** and **deny icmp** commands allows TCP or UDP traffic to be matched to the ULP if an AH is present. TCP or UDP traffic without an AH will not be matched.

IPv6 traffic can be matched to a ULP when an AH header is present. To perform this function, enter the **ahp** option for the *protocol* argument when using the **permit** or **deny** command.

This task shows how to allow TCP or UDP traffic to be matched to the ULP if an AH is present.

SUMMARY STEPS

- enable**
 - configure terminal**
 - ipv6 access-list** *access-list-name*
 - permit icmp auth**
- or
- deny icmp auth**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router# enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 access-list access-list-name Example: Router(config)# ipv6 access-list list1	Defines an IPv6 access list and places the router in IPv6 access list configuration mode.
Step 4	permit icmp auth or deny icmp auth Example: Router(config-ipv6-acl)# permit icmp auth	Specifies permit or deny conditions for an IPv6 ACL using the auth keyword, which is used to match against the presence of the AH.

Creating an IPv6 ACL for Traffic Filtering in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

The following tasks describe how to create and apply ACLs in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

- [Creating an IPv6 ACL in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 12](#)
- [Applying the IPv6 ACL to an Interface in Cisco IOS Release 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases, page 13](#)

Creating an IPv6 ACL in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

This task explains how to create an IPv6 ACL and configure the IPv6 ACL to pass or block traffic in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

Restrictions

- The *source-ipv6-prefix* argument filters traffic by packet source address, and the *destination-ipv6-prefix* argument filters traffic by packet destination address.

- The Cisco IOS software compares an IPv6 prefix against the permit and deny condition statements in the access list. Every IPv6 access list, including access lists that do not have any permit and deny condition statements, has an implicit deny any any statement as its last match condition. The priority or sequence value applied to each condition statement dictates the order in which the statement is applied in the access list.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list** *access-list-name* {**permit** | **deny**} {*source-ipv6-prefix/prefix-length* | **any**} {*destination-ipv6-prefix/prefix-length* | **any**} [**priority** *value*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 access-list <i>access-list-name</i> { permit deny } { <i>source-ipv6-prefix/prefix-length</i> any } { <i>destination-ipv6-prefix/prefix-length</i> any } [priority <i>value</i>] Example: Router(config)# ipv6 access-list list2 deny fec0:0:0:2::/64 any	Creates an IPv6 ACL and sets deny or permit conditions for the ACL.

Applying the IPv6 ACL to an Interface in Cisco IOS Release 12.2(11)T, 12.0(22)S, or Earlier Releases

This task describes how to apply the IPv6 ACL to an interface in Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 traffic-filter** *access-list-name* {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Router(config)# interface ethernet 0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	ipv6 traffic-filter <i>access-list-name {in out}</i> Example: Router(config-if)# ipv6 traffic-filter list2 out	Applies the specified IPv6 access list to the interface specified in the previous step.

Configuring the Cisco IOS Firewall for IPv6

This task shows how to configure the Cisco IOS Firewall for IPv6 environments. This configuration scenario uses both packet inspection and ACLs.

SUMMARY STEPS

- enable**
- configure terminal**
- ipv6 unicast-routing**
- ipv6 inspect name** *inspection-name protocol [alert {on | off}] [audit-trail {on | off}] [timeout seconds]*
- interface** *type number*
- ipv6 address** *{ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}*
- ipv6 enable**
- ipv6 traffic-filter** *access-list-name {in | out}*
- ipv6 inspect** *inspect-name*
- ipv6 access-list** *access-list-name*
- permit** *protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number]*

[sequence value] [time-range name]

or

deny protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [routing] [routing-type routing-number] [sequence value] [time-range name] [undetermined-transport]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables IPv6 unicast routing.
Step 4	ipv6 inspect name inspection-name protocol [alert {on off}] [audit-trail {on off}] [timeout seconds] Example: Router(config)# ipv6 inspect name ipv6_test icmp timeout 60	Defines a set of IPv6 inspection rules for the firewall.
Step 5	interface type number Example: Router(config)# interface FastEthernet0/0	Specifies the interface on which the inspection will occur.
Step 6	ipv6 address {ipv6-address/prefix-length prefix-name sub-bits/prefix-length} Example: Router(config-if)# ipv6 address 3FFE:C000:0:7::/64 eui-64	Provides the address for the inspection interface.
Step 7	ipv6 enable Example: Router(config-if)# ipv6 enable	Enables IPv6 routing. Note This step is optional if the IPv6 address is specified in step 6.

	Command or Action	Purpose
Step 8	ipv6 traffic-filter <i>access-list-name</i> { in out } Example: Router(config-if)# ipv6 traffic-filter outbound out	Applies the specified IPv6 access list to the interface specified in the previous step.
Step 9	ipv6 inspect <i>inspection-name</i> { in out } Example: Router(config)#ipv6 inspect ipv6_test in	Applies the set of inspection rules.
Step 10	ipv6 access-list <i>access-list-name</i> Example: Router(config)# ipv6 access-list outbound	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.
Step 11	permit protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [reflect <i>name</i>] [timeout <i>value</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] or deny protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] [undetermined-transport] Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 any reflect reflectout or Example: Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any	Specifies permit or deny conditions for an IPv6 ACL.

Configuring PAM for IPv6

The tasks in the following sections explain how to configure PAM for IPv6.

- [Creating an IPv6 Access Class Filter for PAM, page 17](#)
- [Applying the IPv6 Access Class Filter to PAM, page 18](#)

Creating an IPv6 Access Class Filter for PAM

The following task explains how to create an IPv6 access class filter to use in PAM configuration:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list access-list-name**
4. **permit protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [reflect name [timeout value]] [routing] [routing-type routing-number] [sequence value] [time-range name]**
or
deny protocol {source-ipv6-prefix/prefix-length | any | host source-ipv6-address | auth} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any | host destination-ipv6-address | auth} [operator [port-number]] [dest-option-type [doh-number | doh-type]] [dscp value] [flow-label value] [fragments] [log] [log-input] [mobility] [mobility-type [mh-number | mh-type]] [routing] [routing-type routing-number] [sequence value] [time-range name] [undetermined-transport]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
	Example: Router> enable	
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

	Command or Action	Purpose
Step 3	ipv6 access-list <i>access-list-name</i> Example: Router(config)# ipv6 access-list outbound	Defines an IPv6 ACL and enters IPv6 access list configuration mode. The router prompt changes to Router(config-ipv6-acl)#.
Step 4	permit protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [reflect <i>name</i>] [timeout <i>value</i>] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] or deny protocol { <i>source-ipv6-prefix/prefix-length</i> any host <i>source-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] { <i>destination-ipv6-prefix/prefix-length</i> any host <i>destination-ipv6-address</i> auth } [<i>operator</i> [<i>port-number</i>]] [dest-option-type [<i>doh-number</i> <i>doh-type</i>]] [dscp <i>value</i>] [flow-label <i>value</i>] [fragments] [log] [log-input] [mobility] [mobility-type [<i>mh-number</i> <i>mh-type</i>]] [routing] [routing-type <i>routing-number</i>] [sequence <i>value</i>] [time-range <i>name</i>] [undetermined-transport] Example: Router(config-ipv6-acl)# permit tcp 2001:0DB8:0300:0201::/32 any reflect reflectout or Example: Router(config-ipv6-acl)# deny tcp fec0:0:0:0201::/64 any	Specifies permit or deny conditions for an IPv6 ACL.

Applying the IPv6 Access Class Filter to PAM

Once you have created an IPv6 access class filter, use the following task to apply the filter to PAM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 port-map** *application-name* **port** *port-num* [**list** *acl-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 port-map <i>application-name</i> port <i>port-num</i> [list <i>acl-name</i>] Example: Router(config)# ipv6 port-map ftp port 8090 list PAMACL	Establishes PAM for the system.

Verifying IPv6 Security Configuration and Operation

This task explains how to display information to verify the configuration and operation of IPv6 security options. Use the following commands as needed to verify configuration and operation.

SUMMARY STEPS

1. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **interface** *interface-type* *interface-number* | **peer** [**vrf** *fvrif-name*] **address** | **vrf** *ivrf-name* | **ipv6** [*interface-type* *interface-number*]] [**detail**]
2. **show crypto isakmp peer** [**config** | **detail**]
3. **show crypto isakmp profile**
4. **show crypto isakmp sa** [**active** | **standby** | **detail** | **nat**]
5. **show ipv6 access-list** [*access-list-name*]
6. **show ipv6 inspect** {**name** *inspection-name* | **config** | **interfaces** | **session** [**detail**] | **all**}
7. **show ipv6 port-map** [*application* | **port** *port-number*]
8. **show ipv6 prefix-list** [**detail** | **summary**] [*list-name*]
9. **show ipv6 virtual-reassembly interface** *interface-type*
10. **show logging** [*slot slot-number* | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	show crypto ipsec sa [map <i>map-name</i> address identity interface <i>interface-type</i> <i>interface-number</i> peer [vrf <i>fvrf-name</i>] address vrf <i>ivrf-name</i> ipv6 [<i>interface-type</i> <i>interface-number</i>]] [detail] Example: Router# show crypto ipsec sa ipv6	Displays the settings used by current SAs.
Step 2	show crypto isakmp peer [config detail] Example: Router# show crypto isakmp peer	Displays peer descriptions.
Step 3	show crypto isakmp profile Example: Router# show crypto isakmp profile	Lists all the ISAKMP profiles that are defined on a router.
Step 4	show crypto isakmp sa [active standby detail nat] Example: Router# show crypto isakmp sa	Displays current IKE SAs.
Step 5	show ipv6 access-list [<i>access-list-name</i>] Example: Router# show ipv6 access-list	Displays the contents of all current IPv6 access lists.
Step 6	show ipv6 inspect { name <i>inspection-name</i> config interfaces session [detail] all } Example: Router# show ipv6 inspect interfaces	Displays CBAC configuration and session information.
Step 7	show ipv6 port-map [<i>application</i> port <i>port-number</i>] Example: Router# show ipv6 port-map ftp	Displays PAM configuration.
Step 8	show ipv6 prefix-list [detail summary] [<i>list-name</i>] Example: Router# show ipv6 prefix-list	Displays information about an IPv6 prefix list or IPv6 prefix list entries.

	Command or Action	Purpose
Step 9	show ipv6 virtual-reassembly interface <i>interface-type</i> Example: Router# show ipv6 virtual-reassembly interface e1/1	Displays configuration and statistical information of VFR.
Step 10	show logging [slot <i>slot-number</i> summary] Example: Router# show logging	Displays the state of system logging (syslog) and the contents of the standard system logging buffer. <ul style="list-style-type: none"> Access list entries with the log or log-input keywords will be logged when a packet matches the access list entry.

Troubleshooting IPv6 Security Configuration and Operation

This optional task explains how to display information to troubleshoot the configuration and operation of IPv6 security options. Use the following commands only as needed to verify configuration and operation.

SUMMARY STEPS

1. **enable**
2. **clear ipv6 access-list** [*access-list-name*]
3. **clear ipv6 inspect** {*session session-number* | **all**}
4. **clear ipv6 prefix-list** [*prefix-list-name*] [*ipv6-prefix/prefix-length*]
5. **debug crypto ipsec**
6. **debug crypto engine packet** [*detail*]
7. **debug ipv6 inspect** {*function-trace* | *object-creation* | *object-deletion* | *events* | *timers* | *protocol* | *detailed*}
8. **debug ipv6 packet** [*access-list access-list-name*] [*detail*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router# enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	clear ipv6 access-list [<i>access-list-name</i>] Example: Router# clear ipv6 access-list tin	Resets the IPv6 access list match counters.

	Command or Action	Purpose
Step 3	clear ipv6 inspect { <i>session session-number</i> <i>all</i> } Example: Router# clear ipv6 inspect all	Removes a specific IPv6 session or all IPv6 inspection sessions.
Step 4	clear ipv6 prefix-list [<i>prefix-list-name</i>] [<i>ipv6-prefix/prefix-length</i>] Example: Router# clear ipv6 prefix-list	Resets the hit count of the IPv6 prefix list entries.
Step 5	debug crypto ipsec Example: Router# debug crypto ipsec	Displays IPsec network events.
Step 6	debug crypto engine packet [<i>detail</i>] Example: Router# debug crypto engine packet	<div>  Caution Using this command could flood the system and increase CPU usage if several packets are being encrypted. </div>
Step 7	debug ipv6 inspect { <i>function-trace</i> <i>object-creation</i> <i>object-deletion</i> <i>events</i> <i>timers</i> <i>protocol</i> <i>detailed</i> } Example: Router# debug ipv6 inspect timers	Displays messages about Cisco IOS Firewall events.
Step 8	debug ipv6 packet [<i>access-list access-list-name</i>] [<i>detail</i>] Example: Router# debug ipv6 packet access-list PAK-ACL	Displays debugging messages for IPv6 packets.

Examples

This section provides the following output examples:

- [Sample Output from the show crypto ipsec sa ipv6 Command, page 23](#)
- [Sample Output from the show crypto isakmp peer Command, page 24](#)
- [Sample Output from the show crypto isakmp profile Command, page 24](#)
- [Sample Output from the show crypto isakmp sa Command, page 24](#)
- [Sample Output from the show ipv6 access-list Command, page 25](#)
- [Sample Output from the show ipv6 prefix-list Command, page 25](#)
- [Sample Output from the show ipv6 virtual-reassembly Command, page 25](#)
- [Sample Output from the show logging Command, page 26](#)
- [Sample Output from the clear ipv6 access-list Command, page 26](#)

Sample Output from the show crypto ipsec sa ipv6 Command

The following is sample output from the **show crypto ipsec sa ipv6** command:

Router# **show crypto ipsec sa ipv6**

```
interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::A8BB:CCFF:FE01:9002

  protected vrf: (none)
  local ident (addr/mask/prot/port): (::/0/0/0)
  remote ident (addr/mask/prot/port): (::/0/0/0)
  current_peer 3FFE:2002::A8BB:CCFF:FE01:2C02 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 133, #pkts encrypt: 133, #pkts digest: 133
    #pkts decaps: 133, #pkts decrypt: 133, #pkts verify: 133
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 60, #recv errors 0

  local crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:9002,
  remote crypto endpt.: 3FFE:2002::A8BB:CCFF:FE01:2C02
  path mtu 1514, ip mtu 1514
  current outbound spi: 0x28551D9A(676666778)

  inbound esp sas:
    spi: 0x2104850C(553944332)
      transform: esp-des ,
      in use settings ={Tunnel, }
      conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
      sa timing: remaining key lifetime (k/sec): (4397507/148)
      IV size: 8 bytes
      replay detection support: Y
      Status: ACTIVE

  inbound ah sas:
    spi: 0x967698CB(2524354763)
      transform: ah-sha-hmac ,
      in use settings ={Tunnel, }
      conn id: 93, flow_id: SW:93, crypto map: Tunnel0-head-0
      sa timing: remaining key lifetime (k/sec): (4397507/147)
      replay detection support: Y
      Status: ACTIVE

  inbound pcp sas:

  outbound esp sas:
    spi: 0x28551D9A(676666778)
      transform: esp-des ,
      in use settings ={Tunnel, }
      conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
      sa timing: remaining key lifetime (k/sec): (4397508/147)
      IV size: 8 bytes
      replay detection support: Y
      Status: ACTIVE

  outbound ah sas:
    spi: 0xA83E05B5(2822636981)
      transform: ah-sha-hmac ,
      in use settings ={Tunnel, }
      conn id: 94, flow_id: SW:94, crypto map: Tunnel0-head-0
      sa timing: remaining key lifetime (k/sec): (4397508/147)
      replay detection support: Y
```

```
Status: ACTIVE
```

```
outbound pcp sas:
```

Sample Output from the show crypto isakmp peer Command

The following sample output shows peer descriptions on an IPv6 router:

```
Router# show crypto isakmp peer detail

Peer: 2001:0DB8:0:1::1 Port: 500 Local: 2001:0DB8:0:2::1
Phase1 id: 2001:0DB8:0:1::1
flags:
NAS Port: 0 (Normal)
IKE SAs: 1 IPsec SA bundles: 1
last_locker: 0x141A188, last_last_locker: 0x0
last_unlocker: 0x0, last_last_unlocker: 0x0
```

Sample Output from the show crypto isakmp profile Command

The following sample output shows the ISAKMP profiles that are defined on an IPv6 router:

```
Router# show crypto isakmp profile

ISAKMP PROFILE tom
Identities matched are:
ipv6-address 2001:0DB8:0:1::1/32
Certificate maps matched are:
Identity presented is: ipv6-address fgdn
keyring(s): <none>
trustpoint(s): <all>
```

Sample Output from the show crypto isakmp sa Command

The following sample output shows the SAs of an active IPv6 device. The IPv4 device is inactive:

```
Router# show crypto isakmp sa detail

Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local          Remote          I-VRF    Status Encr Hash Auth DH
Lifetime Cap.

IPv6 Crypto ISAKMP SA

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1001 I-VRF:          Status: ACTIVE Encr: des Hash: sha Auth:
psk
DH: 1 Lifetime: 23:45:00 Cap: D Engine-id:Conn-id = SW:1

dst: 3FFE:2002::A8BB:CCFF:FE01:2C02
src: 3FFE:2002::A8BB:CCFF:FE01:9002
conn-id: 1002 I-VRF:          Status: ACTIVE Encr: des Hash: sha Auth:
psk
DH: 1 Lifetime: 23:45:01 Cap: D Engine-id:Conn-id = SW:2
```

Sample Output from the show ipv6 access-list Command

In the following example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Router> show ipv6 access-list
```

```
IPv6 access list inbound
  permit tcp any any eq bgp reflect tcptraffic (8 matches) sequence 10
  permit tcp any any eq telnet reflect tcptraffic (15 matches) sequence 20
  permit udp any any reflect udptraffic sequence 30

IPv6 access list tcptraffic (reflexive) (per-user)
  permit tcp host 2001:0DB8:1::32 eq bgp host 2001:0DB8:2::32 eq 11000 timeout 300 (time
    left 243) sequence 1
  permit tcp host 2001:0DB8:1::32 eq telnet host 2001:0DB8:2::32 eq 11001 timeout 300
    (time left 296) sequence 2

IPv6 access list outbound
  evaluate udptraffic
  evaluate tcptraffic
```

Sample Output from the show ipv6 prefix-list Command

The following example shows the output of the **show ipv6 prefix-list** command with the **detail** keyword:

```
Router# show ipv6 prefix-list detail
```

```
Prefix-list with the last deletion/insertion: bgp-in
ipv6 prefix-list 6to4:
  count: 1, range entries: 0, sequences: 5 - 5, refcount: 2
  seq 5 permit 2001:0db8::/32 (hit count: 313, refcount: 1)
ipv6 prefix-list aggregate:
  count: 2, range entries: 2, sequences: 5 - 10, refcount: 30
  seq 5 deny 3FFE:C00::/24 ge 25 (hit count: 568, refcount: 1)
  seq 10 permit ::/0 le 48 (hit count: 31310, refcount: 1)
ipv6 prefix-list bgp-in:
  count: 6, range entries: 3, sequences: 5 - 30, refcount: 31
  seq 5 deny 5F00::/8 le 128 (hit count: 0, refcount: 1)
  seq 10 deny ::/0 (hit count: 0, refcount: 1)
  seq 15 deny ::/1 (hit count: 0, refcount: 1)
  seq 20 deny ::/2 (hit count: 0, refcount: 1)
  seq 25 deny ::/3 ge 4 (hit count: 0, refcount: 1)
  seq 30 permit ::/0 le 128 (hit count: 240664, refcount: 0)
```

Sample Output from the show ipv6 virtual-reassembly Command

The following example shows the output of the **show ipv6 virtual-reassembly** command with the **interface** keyword:

```
Router# show ipv6 virtual-reassembly interface e1/1
```

```
Configuration Information:
```

```
-----
```

```
Virtual Fragment Reassembly (VFR) is ENABLED...
```

```
Maximum number of datagram that can be reassembled at a time: 64
```

```
Maximum number of fragments per datagram: 8
```

```
Timeout value of a datagram: 3 seconds
```

```
Statistical Information:
```

```
-----
```

```
Number of datagram being reassembled:12
```

```
Number of fragments being processed:48
```

```
Total number of datagram reassembled:6950
```

```
Total number of datagram failed: 9
```

Sample Output from the show logging Command

In the following example, the **show logging** command is used to display logging entries that match the first line (sequence 10) of the access list named list1:

```
Router> show logging
```

```
00:00:36: %IPV6-6-ACCESSLOGP: list list1/10 permitted tcp 2001:0db8:1::1(11001)
(Ethernet0/0) -> 2001:0db8:1::2(179), 1 packet
```

Sample Output from the clear ipv6 access-list Command

In the following example, the **show ipv6 access-list** command is used to display some match counters for the access list named list1. The **clear ipv6 access-list** command is issued to reset the match counters for the access list named list1. The **show ipv6 access-list** command is used again to show that the match counters have been reset.

```
Router> show ipv6 access-list list1
```

```
IPv6 access list list1
  permit tcp any any log-input (6 matches) sequence 10
  permit icmp any any echo-request log-input sequence 20
  permit icmp any any echo-reply log-input sequence 30
```

```
Router# clear ipv6 access-list list1
```

```
Router# show ipv6 access-list list1
```

```
IPv6 access list list1
  permit tcp any any log-input sequence 10
  permit icmp any any echo-request log-input sequence 20
  permit icmp any any echo-reply log-input sequence 30
```

Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security

This section provides the following configuration examples:

- [Creating and Applying IPv6 ACLs: Examples, page 26](#)
- [Controlling Access to a vty: Example, page 28](#)
- [Configuring TCP or UDP Matching: Example, page 28](#)
- [Configuring Cisco IOS Firewall for IPv6: Example, page 28](#)

Creating and Applying IPv6 ACLs: Examples

The following sections provide examples for creating and applying ipv6 ACLs:

- [Creating and Applying an IPv6 ACL for Release 12.2\(13\)T or 12.0\(23\)S: Example, page 26](#)
- [Creating and Applying an IPv6 ACL for 12.2\(11\)T, 12.0\(22\)S, or Earlier Releases: Example, page 27](#)

Creating and Applying an IPv6 ACL for Release 12.2(13)T or 12.0(23)S: Example

The following example is from a router running Cisco IOS Release 12.2(13)T.

The example configures two IPv6 ACLs named OUTBOUND and INBOUND and applies both ACLs to outbound and inbound traffic on Ethernet interface 0. The first and second permit entries in the OUTBOUND list permit all TCP and User Datagram Protocol (UDP) packets from network 2001:0DB8:0300:0201::/32 to exit out of Ethernet interface 0. The entries also configure the temporary IPv6 reflexive ACL named REFLECTOUT to filter returning (incoming) TCP and UDP packets on Ethernet interface 0. The first deny entry in the OUTBOUND list keeps all packets from the network fec0:0:0:0201::/64 (packets that have the site-local prefix fec0:0:0:0201 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0.

The **evaluate** command in the INBOUND list applies the temporary IPv6 reflexive ACL named REFLECTOUT to inbound TCP and UDP packets on Ethernet interface 0. When outgoing TCP or UDP packets are permitted on Ethernet interface 0 by the OUTBOUND list, the INBOUND list uses the REFLECTOUT list to match (evaluate) the returning (incoming) TCP and UDP packets.

```
ipv6 access-list OUTBOUND
 permit tcp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 permit udp 2001:0DB8:0300:0201::/32 any reflect REFLECTOUT
 deny fec0:0:0:0201::/64 any

ipv6 access-list INBOUND
 evaluate REFLECTOUT

interface ethernet 0
 ipv6 traffic-filter OUTBOUND out
 ipv6 traffic-filter INBOUND in
```

**Note**

Given that a **permit any any** statement is not included as the last entry in the OUTBOUND or INBOUND ACL, only TCP and UDP packets matching the configured permit entries in the ACL and ICMP packets matching the implicit permit conditions in the ACL are permitted out of and in to Ethernet interface 0 (the implicit deny all condition at the end of the ACL denies all other packet types on the interface).

The following example can be run on a router running Cisco IOS Release 12.2(13)T or 12.0(23)S.

The example configures HTTP access to be restricted to certain hours during the day, and to log any activity outside of the permitted hours:

```
time-range lunchtime
 periodic weekdays 12:00 to 13:00

ipv6 access-list OUTBOUND
 permit tcp any any eq www time-range lunchtime
 deny tcp any any eq www log-input
 permit tcp 2001:0DB8::/32 any
 permit udp 2001:0DB8::/32 any
```

Creating and Applying an IPv6 ACL for 12.2(11)T, 12.0(22)S, or Earlier Releases: Example

The following example is from a router running Cisco IOS Release 12.2(11)T, 12.0(22)S, or earlier releases.

The example configures the IPv6 ACL named list2 and applies the ACL to outbound traffic on Ethernet interface 0. Specifically, the first ACL entry keeps all packets from the network fec0:0:0:2::/64 (packets that have the site-local prefix fec0:0:0:2 as the first 64 bits of their source IPv6 address) from exiting out of Ethernet interface 0. The second entry in the ACL permits all other traffic to exit out of Ethernet interface 0. The second entry is necessary because an implicit deny all condition is at the end of each IPv6 ACL.

```
ipv6 access-list list2 deny fec0:0:0:2::/64 any
```

```
ipv6 access-list list2 permit any any
```

```
interface ethernet 0
  ipv6 traffic-filter list2 out
```

If the same configuration was used on a router running Cisco IOS Release 12.2(13)T, 12.0(23)S, or later releases, the configuration would be translated into IPv6 access list configuration mode as follows:

```
ipv6 access-list list2
  deny ipv6 fec0:0:0:2::/64 any
  permit ipv6 any any
```

```
interface ethernet 0
  ipv6 traffic-filter list2 out
```


Note

IPv6 is automatically configured as the protocol type in **permit any any** and **deny any any** statements that are translated from global configuration mode to IPv6 access list configuration mode.

Controlling Access to a vty: Example

In the following example, incoming connections to the virtual terminal lines 0 to 4 are filtered based on the IPv6 access list named acl1:

```
ipv6 access-list acl1
  permit ipv6 host 2001:0DB8:0:4::2/32 any
  !
line vty 0 4
  ipv6 access-class acl1 in
```

Configuring TCP or UDP Matching: Example

The following example allows any TCP traffic regardless of whether or not an AH is present:

```
IPv6 access list example1
  permit tcp any any
```

The following example allows TCP or UDP parsing only when an AH header is present. TCP or UDP traffic without an AH will not be matched:

```
IPv6 access list example2
  deny tcp host 2001::1 any log sequence 5
  permit tcp any any auth sequence 10
  permit udp any any auth sequence 20
```

The following example allows any IPv6 traffic containing an authentication header:

```
IPv6 access list example3
  permit ahp any any
```

Configuring Cisco IOS Firewall for IPv6: Example

This Cisco IOS Firewall configuration example uses inbound and outbound filters for inspection and makes use of access lists to manage the traffic. The inspect mechanism is the method of permitting return traffic based upon a packet being valid for an existing session for which the state is being maintained:

```
enable
configure terminal
```

```

ipv6 unicast-routing
ipv6 inspect name ipv6_test icmp timeout 60
ipv6 inspect name ipv6_test tcp timeout 60
ipv6 inspect name ipv6_test udp timeout 60

interface FastEthernet0/0
  ipv6 address 3FFE:C000:0:7::/64 eui-64
  ipv6 enable
  ipv6 traffic-filter INBOUND out
  ipv6 inspect ipv6_test in

interface FastEthernet0/1
  ipv6 address 3FFE:C000:1:7::/64 eui-64
  ipv6 enable
  ipv6 traffic-filter OUTBOUND in

! This is used for 3745b connection to tftpboot server
interface FastEthernet4/0
  ip address 192.168.17.33 255.255.255.0
  duplex auto
  speed 100

ip default-gateway 192.168.17.8
! end of tftpboot server config

! Access-lists to deny everything except for Neighbor Discovery ICMP messages
ipv6 access-list INBOUND
  permit icmp any any nd-na
  permit icmp any any nd-ns
  deny ipv6 any any log

ipv6 access-list OUTBOUND
  permit icmp any any nd-na
  permit icmp any any nd-ns
  deny ipv6 any any log

```

Additional References

The following sections provide references related to the Implementing Traffic Filters and Firewalls for IPv6 Security feature.

Related Documents

Related Topic	Document Title
IPv6 IPsec	“Implementing IPsec in IPv6 Security,” Cisco IOS IPv6 Configuration Guide
Basic IPv6 configuration	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2402	<i>IP Authentication Header</i>
RFC 2428	<i>FTP Extensions for IPv6 and NATs</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2474	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>
RFC 3576	<i>Change of Authorization</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **clear ipv6 access-list**
- **clear ipv6 inspect**
- **clear ipv6 prefix-list**
- **debug crypto ipsec**
- **debug crypto engine packet**
- **debug ipv6 inspect**
- **debug ipv6 packet**
- **deny**
- **ipv6 access-class**
- **ipv6 access-list**
- **ipv6 address**
- **ipv6 enable**
- **ipv6 inspect**
- **ipv6 inspect name**
- **ipv6 port-map**
- **ipv6 traffic-filter**
- **ipv6 unicast-routing**
- **permit**
- **show crypto ipsec sa**
- **show crypto isakmp peer**
- **show crypto isakmp profile**
- **show crypto isakmp sa**
- **show ipv6 access-list**
- **show ipv6 inspect**
- **show ipv6 port-map**
- **show ipv6 prefix-list**
- **show ipv6 virtual-reassembly**
- **show logging**

Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security

[Table 1](#) lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifics for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

**Note**

[Table 1](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 **Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security**

Feature Name	Releases	Feature Information
IPv6 services: standard access control lists	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>Access lists determine what traffic is blocked and what traffic is forwarded at router interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2 • Access Control Lists for IPv6 Traffic Filtering, page 2 • PAM in Cisco IOS Firewall for IPv6, page 4 • How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5 • Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26
IPv6 services: extended access control lists ¹	12.0(23)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(13)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	<p>Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Restrictions for Implementing Traffic Filters and Firewalls for IPv6 Security, page 2 • Access Control Lists for IPv6 Traffic Filtering, page 2 • How to Implement Traffic Filters and Firewalls for IPv6 Security, page 5 • Configuration Examples for Implementing Traffic Filters and Firewalls for IPv6 Security, page 26
IPv6 services: IPv6 IOS Firewall	12.3(7)T 12.4 12.4(2)T	<p>This feature provides advanced traffic filtering functionality as an integral part of a network's firewall.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Cisco IOS Firewall for IPv6, page 3 • Configuring the Cisco IOS Firewall for IPv6, page 14

Table 1 **Feature Information for Implementing Traffic Filters and Firewalls for IPv6 Security (continued)**

Feature Name	Releases	Feature Information
IPv6 services: IPv6 IOS Firewall FTP application support	12.3(11)T 12.4 12.4(2)T	IPv6 supports this feature. The following section provides information about this feature: <ul style="list-style-type: none"> • Cisco IOS Firewall for IPv6, page 3
IPv6 ACL extensions for IPsec Authentication Header	12.4(20)T	The IPv6 ACL extensions for IPsec authentication headers feature allows TCP or UDP parsing when an IPv6 IPsec authentication header is present. The following section provides information about this feature: <ul style="list-style-type: none"> • IPv6 ACL Extensions for IPsec Authentication Header, page 2

1. IPv6 extended access control lists and IPv6 provider edge router over Multiprotocol Label Switching (MPLS) are implemented with hardware acceleration on the Cisco 12000 series Internet router IP service engineer (ISE) line cards in Cisco IOS routers in Cisco IOS Release 12.0(25)S and later releases.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing Selective Packet Discard in IPv6

First Published: August 21, 2007

Last Updated: February 25, 2009

This document describes the Selective Packet Discard (SPD) feature in IPv6. The SPD feature in IPv6 manages the process level input queues on the Route Processor (RP). SPD provides priority to routing protocol packets and other important traffic control Layer 2 keepalives during periods of process level queue congestion.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Selective Packet Discard in IPv6”](#) section on [page 7](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

-
- [How to Implement Selective Packet Discard in IPv6, page 3](#)
[Configuration Examples for Implementing Selective Packet Discard in IPv6, page 5](#)
[Additional References, page 5](#)
[Command Reference, page 6](#)
[Feature Information for Implementing Selective Packet Discard in IPv6, page 7](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006–2009 Cisco Systems, Inc. All rights reserved.

Information About Implementing Selective Packet Discard in IPv6

.

SPD in IPv6 Overview

SPD State Check

The SPD state check is performed on the IPv6 process input queue on the RP. High-priority packets, such as those of IP precedence 7, are not applied to SPD and are never dropped. All remaining packets, however, can be dropped depending on the length of the IPv6 packet input queue and the SPD state. The possible SPD states are as follows:

- Normal: The queue size is less than the maximum.

Full drop: The queue size is greater than or equal to the maximum.

In the normal state, the router never drops well-formed and malformed packets. In the full drop state, the router drops all well-formed and malformed packets.

SPD Headroom

With SPD, the behavior of normal IPv6 packets is not changed. However, routing protocol packets are given higher priority, because SPD recognizes routing protocol packets by the IPv6 precedence field. Therefore, if the IPv6 precedence is set to 7, then the packet is given priority.

SPD prioritizes IPv6 packets with a precedence of 7 by allowing the Cisco IOS software to queue them into the process level input queue above the normal input queue limit. The number of packets allowed in excess of the normal limit is called the SPD headroom. The SPD headroom default is 100, which means that a high precedence packet is not dropped if the size of the input hold queue is lower than 175 (which is the input queue default size + SPD headroom size).

Non-IPv6 packets such as Connectionless Network Service Intermediate System-to-Intermediate System (CLNS IS-IS) packets, PPP packets, and High-Level Data Link Control (HDLC) keepalives were treated as normal priority as a result of being Layer 2 instead of Layer 3. In addition, Interior Gateway Protocols (IGPs) operating at Layer 3 or higher are given priority over normal IPv6 packets, but are given the same priority as Border Gateway Protocol (BGP) packets. So, during BGP convergence or during times of very high BGP activity, IGP hellos and keepalives often were dropped, causing IGP adjacencies to fail.

Because IGP and link stability are tenuous and crucial, such packets are given the highest priority and are given extended SPD headroom with a default of 10 packets. These packets are not dropped if the size of the input hold queue is lower than 185 (input queue default size + SPD headroom size + SPD extended headroom).

How to Implement Selective Packet Discard in IPv6

- [Configuring the SPD Process Input Queue, page 3](#) (required)
- [Configuring SPD Headroom, page 4](#) (optional)

Configuring the SPD Process Input Queue

SUMMARY STEPS

1. `enable`
`configure terminal`
3. `ipv6 spd queue max-threshold value`
- 4.
5. `show ipv6 spd`

DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>	Enables privileged EXEC mode. Enter your password if prompted.
	Example: <code>Router> enable</code>	
	<code>configure terminal</code>	
	<code>Router# configure terminal</code>	
	<code>ipv6 spd queue max-threshold value</code>	
	<code>Router(config)# ipv6 spd queue max-threshold 100</code>	
	<code>Router(config)# exit</code>	
	<code>show ipv6 spd</code>	
	<code>Router# show ipv6 spd</code>	

Configuring SPD Headroom

SUMMARY STEPS

- 1.
- 2.
- 3.
- 4. *size*
- 5.
- 6.

DETAILED STEPS

	Command or Action	Purpose
Step 1	Example:	<ul style="list-style-type: none">•
Step 2	Example:	
Step 3	<i>size</i> Router(config)# spd headroom 200	
	<i>size</i> Router(config)# spd extended 11	
	Router(config)# exit	
	Router# show ipv6 spd	

Configuration Examples for Implementing Selective Packet Discard in IPv6

.

Configuring the SPD Process Input Queue: Example

```
ipv6 spd queue maximum-threshold 1
show ipv6 spd
```

```
Current mode: normal
Queue max threshold: 1, Headroom: 100, Extended Headroom: 10
IPv6 packet queue: 0
```

Additional References

Related Documents

Related Topic	Document Title
	<i>Cisco IOS IPv6 Configuration Guide</i>
	Cisco IOS IPv6 Command Reference

Standards

Standard	Title
	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
	<i>Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers</i>

Technical Assistance

Command Reference

http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*

[Start Here:](#)

[Cisco IOS Software Release Specifies for IPv6 Features](#)



Note

Table 1 *Feature Information for Implementing Selective Packet Discard in IPv6*

Feature Name	Releases	Feature Information

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2007–2009 Cisco Systems, Inc. All rights reserved.



Implementing Static Routes for IPv6

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure static routes for IPv6. Routing defines the paths over which packets travel in the network. Manually configured static routes may be used instead of dynamic routing protocols for smaller networks or for sections of a network that have only one path to an outside network. Lack of redundancy limits the usefulness of static routes, and in larger networks manual reconfiguration of routes can become a large administrative overhead.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Static Routes for IPv6” section on page 17](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing Static Routes for IPv6, page 2](#)
- [Information About Implementing Static Routes for IPv6, page 2](#)
- [How to Implement Static Routes for IPv6, page 4](#)
- [Configuration Examples for Implementing Static Routes for IPv6, page 12](#)
- [Additional References, page 15](#)
- [Command Reference, page 16](#)
- [Feature Information for Implementing Static Routes for IPv6, page 17](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2001–2009 Cisco Systems, Inc. All rights reserved.

Prerequisites for Implementing Static Routes for IPv6

- This document assumes that you are familiar with IPv4. Refer to the publications referenced in the [“Related Documents”](#) section for IPv4 configuration and command reference information.
- Before configuring the router with a static IPv6 route you must enable the forwarding of IPv6 packets using the **ipv6 unicast-routing** global configuration command, enable IPv6 on at least one interface, and configure an IPv6 address on that interface. For details on basic IPv6 connectivity tasks, refer to the [Implementing Ipv6 Addressing and Basic Connectivity](#) module.

Information About Implementing Static Routes for IPv6

To configure static routes for IPv6, you need to understand the following concepts:

- [Static Routes, page 2](#)
- [Directly Attached Static Routes, page 2](#)
- [Recursive Static Routes, page 3](#)
- [Fully Specified Static Routes, page 4](#)
- [Floating Static Routes, page 4](#)

Static Routes

Networking devices forward packets using route information that is either manually configured or dynamically learned using a routing protocol. Static routes are manually configured and define an explicit path between two networking devices. Unlike a dynamic routing protocol, static routes are not automatically updated and must be manually reconfigured if the network topology changes. The benefits of using static routes include security and resource efficiency. Static routes use less bandwidth than dynamic routing protocols and no CPU cycles are used to calculate and communicate routes. The main disadvantage to using static routes is the lack of automatic reconfiguration if the network topology changes.

Static routes can be redistributed into dynamic routing protocols but routes generated by dynamic routing protocols cannot be redistributed into the static routing table. No algorithm exists to prevent the configuration of routing loops that use static routes.

Static routes are useful for smaller networks with only one path to an outside network and to provide security for a larger network for certain types of traffic or links to other networks that need more control. In general, most networks use dynamic routing protocols to communicate between networking devices but may have one or two static routes configured for special cases.

Directly Attached Static Routes

In directly attached static routes, only the output interface is specified. The destination is assumed to be directly attached to this interface, so the packet destination is used as the next-hop address. This example shows such a definition:

```
ipv6 route 2001:0DB8::/32 ethernet1/0
```

The example specifies that all destinations with address prefix 2001:0DB8::/32 are directly reachable through interface Ethernet1/0.

Directly attached static routes are candidates for insertion in the IPv6 routing table only if they refer to a valid IPv6 interface; that is, an interface that is both up and has IPv6 enabled on it.

Recursive Static Routes

In a recursive static route, only the next hop is specified. The output interface is derived from the next hop. This example shows such a definition:

```
ipv6 route 2001:0DB8::/32 2001:0DB8:3000:1
```

This example specifies that all destinations with address prefix 2001:0DB8::/32 are reachable via the host with address 2001:0DB8:3000:1.

A recursive static route is valid (that is, it is a candidate for insertion in the IPv6 routing table) only when the specified next hop resolves, either directly or indirectly, to a valid IPv6 output interface, provided the route does not self-recurse, and the recursion depth does not exceed the maximum IPv6 forwarding recursion depth.

A route self-recurses if it is itself used to resolve its own next hop. For example, suppose we have the following routes in the IPv6 routing table:

```
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
R   2001:0DB8::/32 [130/0]
    via ::, Serial2/0
B   2001:0DB8:3000:0/16 [200/45]
    Via 2001:0DB8::0104
```

The following examples defines a recursive IPv6 static route:

```
ipv6 route
2001:0DB8::/32 2001:0DB8:3000:1
```

This static route will not be inserted into the IPv6 routing table because it is self-recursive. The next hop of the static route, 2001:0DB8:3000:1, resolves via the BGP route 2001:0DB8:3000:0/16, which is itself a recursive route (that is, it only specifies a next hop). The next hop of the BGP route, 2001:0DB8::0104, resolves via the static route. Therefore, the static route would be used to resolve its own next hop.

It is not normally useful to manually configure a self-recursive static route, although it is not prohibited. However, a recursive static route that has been inserted in the IPv6 routing table may become self-recursive as a result of some transient change in the network learned through a dynamic routing protocol. If this occurs, the fact that the static route has become self-recursive will be detected and it will be removed from the IPv6 routing table, although not from the configuration. A subsequent network change may cause the static route to no longer be self-recursive, in which case it will be reinserted in the IPv6 routing table.

IPv6 recursive static routes are checked at one-minute intervals. So, a recursive static route may take up to a minute to be inserted into the routing table once its next hop becomes valid. Likewise, it may take a minute or so for the route to disappear from the table if its next hop becomes invalid.

Fully Specified Static Routes

In a fully specified static route, both the output interface and the next hop are specified. This form of static route is used when the output interface is a multi-access one and it is necessary to explicitly identify the next hop. The next hop must be directly attached to the specified output interface. The following example shows a definition of a fully specified static route:

```
ipv6 route 2001:DB8:/32 ethernet1/0 2001:0DB8:3000:1
```

A fully specified route is valid (that is, a candidate for insertion into the IPv6 routing table) when the specified IPv6 interface is IPv6-enabled and up.

Floating Static Routes

Floating static routes are static routes that are used to back up dynamic routes learned through configured routing protocols. A floating static route is configured with a higher administrative distance than the dynamic routing protocol it is backing up. As a result, the dynamic route learned through the routing protocol is always used in preference to the floating static route. If the dynamic route learned through the routing protocol is lost, the floating static route will be used in its place. The following example defines a floating static route:

```
ipv6 route 2001:DB8:/32 ethernet1/0 2001:0DB8:3000:1 210
```

Any of the three types of IPv6 static routes can be used as a floating static route. A floating static route must be configured with an administrative distance that is greater than the administrative distance of the dynamic routing protocol, because routes with smaller administrative distances are preferred.

**Note**

By default, static routes have smaller administrative distances than dynamic routes, so static routes will be used in preference to dynamic routes.

How to Implement Static Routes for IPv6

The following sections explain how to configure static IPv6 routes:

- [Configuring a Static IPv6 Route, page 4](#)
- [Configuring a Floating Static IPv6 Route: Example, page 6](#)
- [Verifying Static IPv6 Route Configuration and Operation, page 7](#)

Configuring a Static IPv6 Route

This task explains how to configure a static default IPv6 route, a static IPv6 route through a point-to-point interface, and a static IPv6 route to a multiaccess interface.

Static Routes in IPv6

Use the **ipv6 route** command to configure IPv6 static routes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix/prefix-length* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>]} [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [tag <i>tag</i>] Example: Router(config)# ipv6 route ::/0 serial 2/0	Configures a static IPv6 route. <ul style="list-style-type: none">• A static default IPv6 route is being configured on a serial interface.• See the syntax examples that immediately follow this table for specific uses of the ipv6 route command for configuring static routes.

Examples

In addition to the syntax example included in the [“DETAILED STEPS” section on page 5](#), the following syntax examples illustrate use of the **ipv6 route** for configuring the various types of static routes.

Directly Attached Static Route Through Point-to-Point Interface Example Syntax

The following example shows how to configure a directly attached static route through a point-to-point interface.

```
Router(config)# ipv6 route 2001:0DB8::/32 serial 0
```

Directly Attached Static Route on Broadcast Interface Example Syntax

The following example shows how to configure a directly attached static route on a broadcast interface.

```
Router(config)# ipv6 route 2001:0DB8::1/32 ethernet1/0
```

Fully Specified Static Route on Broadcast Interface Example Syntax

The following example shows how to configure a fully specified static route on a broadcast interface.

```
Router(config)# ipv6 route 2001:0DB8::1/32 ethernet1/0 fe80::1
```

Recursive Static Route

In the following example, a static route is being configured to a specified next-hop address, from which the output interface is automatically derived.

```
Router(config)# ipv6 route 2001:0DB8::/32 2001:0DB8:2002:1
```

Configuring a Floating Static IPv6 Route: Example

This task explains how to configure a floating static IPv6 route.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 route** *ipv6-prefix/prefix-length* { *ipv6-address* | *interface-type interface-number* [*ipv6-address*] } [*administrative-distance*] [*administrative-multicast-distance* | **unicast** | **multicast**] [**tag** *tag*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 route <i>ipv6-prefix/prefix-length</i> { <i>ipv6-address</i> <i>interface-type interface-number</i> [<i>ipv6-address</i>] } [<i>administrative-distance</i>] [<i>administrative-multicast-distance</i> unicast multicast] [<i>tag tag</i>] Example: Router(config)# ipv6 route 2001:0DB8::/32 serial 2/0 201	Configures a static IPv6 route. <ul style="list-style-type: none"> In this example, a floating static IPv6 route is being configured. An administrative distance of 200 is configured. Default administrative distances are as follows: <ul style="list-style-type: none"> Connected interface—0 Static route—1 Enhanced Interior Gateway Routing Protocol (EIGRP) summary route—5 External Border Gateway Protocol (eBGP)—20 Internal Enhanced IGRP—90 IGRP—100 Open Shortest Path First—110 Intermediate System-to-Intermediate System (IS-IS)—115 Routing Information Protocol (RIP)—120 Exterior Gateway Protocol (EGP)—140 EIGRP external route—170 Internal BGP—200 Unknown—255

Verifying Static IPv6 Route Configuration and Operation

This task explains how to display information to verify the configuration and operation of static IPv6 routes.

Use the **show ipv6 static** command to display a set of static routes and the installed status of each, that is, whether an entry for each route appears in the IPv6 routing table.

Use the **show ipv6 route** command to confirm that installed routes are in the IPv6 routing table and that each route definition reflects the expected cost and metric. If a static route that you have configured does not appear in the IPv6 routing table, it is possible that there is a lower administrative distance from another source in the table, such as from a routing protocol. Such a change to the routing table would occur only if you have specified a non-default administrative distance on the static route.

If a lower administrative distance exists, the static route is “floating” and will be inserted into the routing table only when the route learned through the routing protocol disappears. If there is not a lower administrative distance in the routing table, then the static route should be used.

Use the **show ipv6 static** command with the **detail** keyword to determine what is causing any discrepancy. For example, if the static route is a direct static route, the interface might be down or IPv6 might not be enabled on the interface.

SUMMARY STEPS

1. **enable**
2. **show ipv6 static** [*ipv6-address* | *ipv6-prefix/prefix-length*][**interface** *interface-type* *interface-number*] [**recursive**] [**detail**]
or
show ipv6 route [*ipv6-address* | *ipv6-prefix/prefix-length* | *protocol* | *interface-type* *interface-number*]
3. **debug ipv6 routing**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show ipv6 static [<i>ipv6-address</i> <i>ipv6-prefix/prefix-length</i>][interface <i>interface-type interface-number</i>] [recursive] [detail] or show ipv6 route [<i>ipv6-address</i> <i>ipv6-prefix/prefix-length</i> <i>protocol</i> <i>interface-type interface-number</i>] Example: Router# show ipv6 static or Example: Router# show ipv6 route static	Displays the current contents of the IPv6 routing table. <ul style="list-style-type: none"> These examples show two different ways of displaying IPv6 static routes.
Step 3	debug ipv6 routing Example: Router# debug ipv6 routing	Displays debugging messages for IPv6 routing table updates and route cache updates.

Examples

This section provides the following output examples:

- [Sample Output from the show ipv6 static Command when No Options Are Specified in the Command Syntax, page 9](#)
- [Sample Output from the show ipv6 static Command with the IPv6 Address and Prefix Command, page 10](#)
- [Sample Output from the show ipv6 static interface Command, page 10](#)
- [Sample Output from the show ipv6 static recursive Command, page 10](#)
- [Sample Output from the show ipv6 static detail Command, page 10](#)
- [Sample Output from the show ipv6 route Command, page 11](#)
- [Sample Output from the debug ipv6 routing Command, page 11](#)

Sample Output from the show ipv6 static Command when No Options Are Specified in the Command Syntax

When no options are specified in the command, those routes installed in the IPv6 routing table are marked with an asterisk, as shown in the following example:

```
Router# show ipv6 static
```

```
IPv6 Static routes
```

```

Code: * - installed in RIB
* 2001:0DB8:3000:0/16, interface Ethernet1/0, distance 1
* 2001:0DB8:4000:0/16, via nexthop 2001:0DB8:1:1, distance 1
  2001:0DB8:5000:0/16, interface Ethernet3/0, distance 1
* 2001:0DB8:5555:0/16, via nexthop 2001:0DB8:4000:1, distance 1
  2001:0DB8:5555:0/16, via nexthop 2001:0DB8:9999:1, distance 1
* 2001:0DB8:5555:0/16, interface Ethernet2/0, distance 1
* 2001:0DB8:6000:0/16, via nexthop 2001:0DB8:2007:1, interface Ethernet1/0, distance 1

```

Sample Output from the show ipv6 static Command with the IPv6 Address and Prefix Command

When the *ipv6-address* or *ipv6-prefix/prefix-length* argument is specified, only information about static routes for that address or network is displayed. The following is sample output from the **show ipv6 static** command when entered with the IPv6 prefix 2001:0DB8:200::/35:

```
Router# show ipv6 static 2001:0DB8:5555:0/16
```

```

IPv6 Static routes
Code: * - installed in RIB
* 2001:0DB8:5555:0/16, via nexthop 2001:0DB8:4000:1, distance 1
  2001:0DB8:5555:0/16, via nexthop 2001:9999:1, distance 2
* 2001:0DB8:5555:0/16, interface Ethernet2/0, distance 1

```

Sample Output from the show ipv6 static interface Command

When an interface is supplied, only those static routes with the specified interface as outgoing interface are displayed. The **interface** keyword may be used with or without the IPv6 address and prefix specified in the **show ipv6 static** command.

```
Router# show ipv6 static interface ethernet3/0
```

```

IPv6 Static routes
Code: * - installed in RIB

```

Sample Output from the show ipv6 static recursive Command

When the **recursive** keyword is specified in the **show ipv6 static** command, only recursive static routes are displayed. The **recursive** keyword is mutually exclusive with the **interface** keyword, but it may be used *with* or *without* the IPv6 prefix included in the command syntax.

```
Router# show ipv6 static recursive
```

```

IPv6 Static routes
Code: * - installed in RIB
* 2001:0DB8:4000:0/16, via nexthop 2001:0DB8:1:1, distance 1
* 2001:0DB8:5555:0/16, via nexthop 2001:0DB8:4000:1, distance 2
  2001:0DB8:5555:0/16, via nexthop 2001:0DB8:9999:1, distance 3

```

Sample Output from the show ipv6 static detail Command

When the **detail** keyword is specified, the following additional information is also displayed:

- For *valid* recursive routes, the output path set, and maximum resolution depth
- For *invalid* recursive routes, the reason why the route is not valid.
- For *invalid* direct or fully-specified routes, the reason why the route is not valid.

```
Router# show ipv6 static detail
```

```

IPv6 Static routes
Code: * - installed in RIB
* 2001:0DB8:3000:0/16, interface Ethernet1/0, distance 1
* 2001:0DB8:4000:0/16, via nexthop 2001:0DB8:2001:1, distance 1
  Resolves to 1 paths (max depth 1)
  via Ethernet1/0

```

```

2001:0DB8:5000:0/16, interface Ethernet3/0, distance 1
Interface is down
* 2001:0DB8:5555:0/16, via nexthop 2001:0DB8:4000:1, distance 1
Resolves to 1 paths (max depth 2)
via Ethernet1/0
2001:0DB8:5555:0/16, via nexthop 2001:0DB8:9999:1, distance 1
Route does not fully resolve
* 2001:0DB8:5555:0/16, interface Ethernet2/0, distance 1
* 2001:0DB8:6000:0/16, via nexthop 2001:0DB8:2007:1, interface Ethernet1/0, distance 1

```

Sample Output from the show ipv6 route Command

In the following example, the **show ipv6 route** command is used to verify the configuration of a static route through a point-to-point interface:

```

Router# show ipv6 route

IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
S   2001:0DB8::/32 [1/0]
    via ::, Serial2/0

```

In the following example, the **show ipv6 route** command is used to verify the configuration of a static route on a multiaccess interface. An IPv6 link-local address—FE80::1—is the next-hop router.

```

Router# show ipv6 route

IPv6 Routing Table - 11 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
S   2001:0DB8::/32 [1/0]
    via FE80::1, Ethernet0/0

```

To display all static routes in the IPv6 routing table, use the **show ipv6 route static** command is used with static as the value of the protocol argument:

```

Router# show ipv6 route static

IPv6 Routing Table - 330 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
S   2001:0DB8::/32 [1/0]
    via ::, Tunnel0
S   3FFE:C00:8011::/48 [1/0]
    via ::, Null0
S   ::/0 [254/0]
    via 2001:0DB8:2002:806B, Null

```

Sample Output from the debug ipv6 routing Command

In the following example, the **debug ipv6 routing** command is used to verify the installation of a floating static route into the IPv6 routing table when an IPv6 RIP route is deleted. The floating static IPv6 route was previously configured with an administrative distance value of 130. The backup route was added as a floating static route because RIP routes have a default administrative distance of 120, and the RIP route should be the preferred route. When the RIP route is deleted, the floating static route is installed in the IPv6 routing table.

```
Router# debug ipv6 routing

*Oct 10 18:28:00.847: IPv6RT0: rip two, Delete 2001:0DB8::/32 from table
*Oct 10 18:28:00.847: IPv6RT0: static, Backup call for 2001:0DB8::/32
*Oct 10 18:28:00.847: IPv6RT0: static, Add 2001:0DB8::/32 to table
*Oct 10 18:28:00.847: IPv6RT0: static, Adding next-hop :: over Serial2/0 for
2001:0DB8::/32, [130/0]
```

Configuration Examples for Implementing Static Routes for IPv6

Static routes may be used for a variety of purposes. Common usages include the following:

- Manual summarization
- Traffic discard
- Fixed default route
- Backup route

In many cases, alternative mechanisms exist within Cisco IOS software to achieve the same objective. Whether to use static routes or one of the alternative mechanisms depends on local circumstances.

This section provides the following configuration examples:

- [Configuring Manual Summarization: Example, page 12](#)
- [Configuring Traffic Discard: Example, page 13](#)
- [Configuring a Fixed Default Route: Example, page 13](#)
- [Configuring a Floating Static IPv6 Route: Example, page 6](#)

Configuring Manual Summarization: Example

The following example shows a static route being used to summarize local interface prefixes advertised into RIP. The static route also serves as a discard route, discarding any packets received by the router to a 2001:0DB8:1::/48 destination not covered by a more specific interface prefix.

```
Router> enable
Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)# interface ethernet0/0
Router(config-if)# ipv6 address 2001:0DB8:2:1234/64
Router(config-if)# exit
Router(config)#
Router(config)# interface ethernet1/0
Router(config-if)# ipv6 address 2001:0DB8:3:1234/64
Router(config-if)# exit
Router(config)#

Router(config)# interface ethernet2/0
Router(config-if)# ipv6 address 2001:0DB8:4:1234/64
Router(config-if)# exit
Router(config)#

Router(config)# interface ethernet3/0
Router(config-if)# ipv6 address 2001:0DB8::1234/64
Router(config-if)# ipv6 rip one enable
```



```

Router(config-if)# exit
Router(config)#

Router(config)# ipv6 router rip one
Router(config-rtr)# redistribute static
Router(config-rtr)# exit
Router(config)#

Router(config)# ipv6 route 2001:0DB8:1:1/48 null0
Router(config)# end
Router#

00:01:30: %SYS-5-CONFIG_I: Configured from console by console

Router# show ipv6 route static

IPv6 Routing Table - 3 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user Static route
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   2001:0DB8:1::/48 [1/0]
    via ::, Null0

```

Configuring Traffic Discard: Example

Configuring a static route to point at interface null0 may be used for discarding traffic to a particular prefix. For example, if it is required to discard all traffic to prefix 2001:0DB8:42:1/64, the following static route would be defined:

```

Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# ipv6 route 2001:0DB8:42:1::/64 null0
Router(config)# end
Router#

00:05:44: %SYS-5-CONFIG_I: Configured from console by console

```

Configuring a Fixed Default Route: Example

A default static route is often used in simple router topologies. In the following example, a router is connected to its local site via Ethernet0/0 and to the main corporate network via Serial2/0 and Serial3/0. All nonlocal traffic will be routed over the two serial interfaces.

```

Router(config)# interface ethernet0/0
Router(config-if)# ipv6 address 2001:0DB8:17:1234/64
Router(config-if)# exit

Router(config)# interface Serial2/0
Router(config-if)# ipv6 address 2001:0DB8:1:1234/64
Router(config-if)# exit

Router(config)# interface Serial3/0
Router(config-if)# ipv6 address 2001:0DB8:2:124/64
Router(config-if)# exit

Router(config)# ipv6 route ::0 Serial2/0

```

```

Router(config)# ipv6 route ::/0 Serial13/0
Router(config)# end
Router#

00:06:30: %SYS-5-CONFIG_I: Configured from console by console
Router# show ipv6 route static

IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S    ::/0 [1/0]
    via ::, Serial2/0
    via ::, Serial3/0

```

Configuring a Floating Static Route: Example

A floating static route often is used to provide a backup path in the event of connectivity failure. In the following example, the router has connectivity to the network core via Serial2/0 and learns the route 2001:0DB8:1:1/32 via IS-IS. If the Serial2/0 interface fails, or if route 2001:0DB8:1:1/32 is no longer learned via IS-IS (indicating loss of connectivity elsewhere in the network), traffic is routed via the backup ISDN interface.

```

Router> enable
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Router(config)# interface ethernet0/0
Router(config-if)# ipv6 address 2001:0DB8:17:1234/64
Router(config-if)# exit

Router(config)# interface Serial2/0
Router(config-if)# ipv6 address 2001:0DB8:1:1234/64
Router(config-if)# ipv6 router isis
Router(config-if)# exit

Router(config)# router isis
Router(config-router)# net 42.0000.0000.0000.0001.00
Router(config-router)# exit

Router(config)# interface BRI1/0
Router(config-if)# encapsulation ppp
Router(config-if)# ipv6 enable
Router(config-if)# isdn switch-type basic-net3
Router(config-if)# ppp authentication chap optional
Router(config-if)# ppp multilink
Router(config-if)# exit

Router(config)# dialer-list 1 protocol ipv6 permit
Router(config)# ipv6 route 2001:0DB8:1::/32 BRI1/0 200
Router(config)# end
Router#

00:03:07: %SYS-5-CONFIG_I: Configured from console by console
2001:0DB8:5000:)/16, interface Ethernet3/0, distance 1

```

Where to Go Next

If you want to implement routing protocols, refer to the [Implementing RIP for IPv6](#), [Implementing IS-IS for IPv6](#), [Implementing OSPF for IPv6](#), or [Implementing Multiprotocol BGP for IPv6](#) module.

Additional References

The following sections provide references related to the Implementing Static Routes for IPv6 feature.

Related Documents

Related Topic	Document Title
IP static route configuration	“Protocol-Independent Routing,” Cisco IOS IP Routing Protocols Configuration Guide
IP static route commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IP Routing Protocols Command Reference
IPv6 supported feature list	“Start Here: Cisco IOS Software Release Specifics for IPv6 Features,” Cisco IOS IPv6 Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	—

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **debug ipv6 route**
- **ipv6 route**
- **ipv6 route static bfd**
- **monitor event ipv6 static**
- **show ipv6 route**
- **show ipv6 route summary**
- **show ipv6 static**

Feature Information for Implementing Static Routes for IPv6

Table 13 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information about a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 13 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 13 **Feature Information for Implementing Static Routes for IPv6**

Feature Name	Releases	Feature Information
IPv6 routing: static routing	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Static routes are manually configured and define an explicit path between two networking devices. This entire document provides information about this feature.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.



Implementing VoIP for IPv6

First Published: October 10, 2008

Last Updated: October 10, 2008

This document describes VoIP in IPv6 (VoIPv6), a feature that adds IPv6 capability to existing VoIP features. This feature adds dual-stack (IPv4 and IPv6) support on voice gateways and media termination points (MTPs), IPv6 support for Session Initiation Protocol (SIP) trunks, and Skinny Client Control Protocol (SCCP)-controlled analog voice gateways. In addition, the Session Border Controller (SBC) functionality of connecting SIP IPv4 or H.323 IPv4 network to SIP IPv6 network is implemented on a Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing VoIP for IPv6” section on page 25](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Prerequisites for Implementing VoIP for IPv6, page 2](#)
- [Restrictions for Implementing VoIP for IPv6, page 2](#)
- [Information About Implementing VoIP for IPv6, page 2](#)
- [How to Implement VoIP for IPv6, page 3](#)
- [Configuration Examples for Implementing VoIP over IPv6, page 19](#)
- [Additional References, page 22](#)
- [Feature Information for Implementing VoIP for IPv6, page 25](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Implementing VoIP for IPv6

- This document assumes that you are familiar with IPv6 and IPv4. See the publications referenced in the [“Additional References”](#) section for IPv6 and IPv4 configuration and command reference information.
- Perform basic IPv6 addressing and basic connectivity as described in [Implementing IPv6 Addressing and Basic Connectivity](#).
- Cisco Express Forwarding for IPv6 must be enabled.
- Perform basic voice configurations as described in the [Voice Configuration Library](#).

Restrictions for Implementing VoIP for IPv6

- The following platforms are supported in Cisco IOS Release 12.4(22)T:
 - Integrated Services Routers (2801, 2821, 2851, 3825, 3845)
 - VG202/204 (Orbity)
 - VG224
 - IAD2430
 - AS5400XM

Information About Implementing VoIP for IPv6

The VoIPv6 feature includes IPv4 and IPv6 dual stack support on voice gateways and MTP, IPv6 support for SIP trunks, and Skinny Client Control Protocol (SCCP)-controlled analog gateways. In addition, connecting SIP IPv4 or H.323 IPv4 network to a SIP IPv6 network is implemented on a Cisco Unified Border Element.

To configure VoIP for IPv6, you should understand the following concepts:

- [SIP Voice Gateways in VoIPv6, page 2](#)
- [Cisco Unified Border Element in VoIPv6, page 3](#)
- [MTP Used with Voice Gateways in VoIPv6, page 3](#)

SIP Voice Gateways in VoIPv6

SIP is a simple, ASCII-based protocol that uses requests and responses to establish communication among the various components in the network and to ultimately establish a conference between two or more endpoints.

For further information about this feature and information about configuring the SIP voice gateway for VoIPv6, see the [“Configuring a SIP Voice Gateway for IPv6”](#) section on page 3.

Cisco Unified Border Element in VoIPv6

The Cisco Unified Border Element feature adds IPv6 capability to existing VoIP features. This feature adds dual-stack support on voice gateways and MTP, IPv6 support for SIP trunks, and SCCP-controlled analog voice gateways.

For further information about this feature and information about configuring the Cisco Unified Border Element in VoIPv6, see the [“Configuring H.323 IPv4-to-SIPv6 Connections in a Cisco Unified Border Element”](#) section on page 15.

MTP Used with Voice Gateways in VoIPv6

Cisco IOS MTP trusted relay point (TRP) supports media interoperation between IPv4 and IPv6 networks.

For further information about this feature and information about configuring the SIP voice gateway for IPv6, see the [“Configuring MTP Used with Voice Gateways”](#) section on page 17.

How to Implement VoIP for IPv6

The following section describes VoIPv6 concepts and how to configure VoIPv6 on the network:

- [Configuring a SIP Voice Gateway for IPv6, page 3](#)
- [Configuring H.323 IPv4-to-SIPv6 Connections in a Cisco Unified Border Element, page 15](#)
- [Configuring MTP Used with Voice Gateways, page 17](#)

Configuring a SIP Voice Gateway for IPv6

SIP is a simple, ASCII-based protocol that uses requests and responses to establish communication among the various components in the network and to ultimately establish a conference between two or more endpoints.

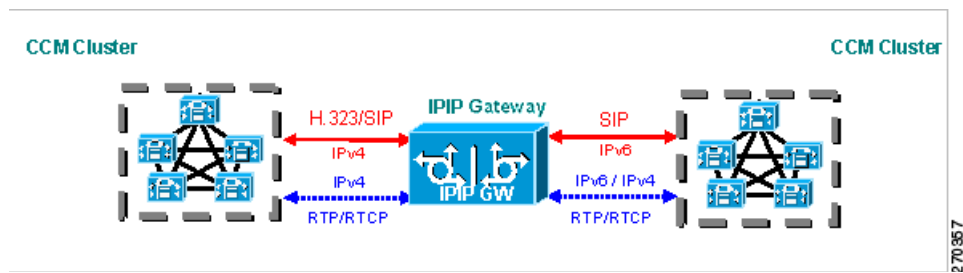
Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and is in the format of sip:userID@gateway.com. The user ID can be either a username or an E.164 address. The gateway can be either a domain (with or without a hostname) or a specific Internet IPv4 or IPv6 address.

A SIP trunk can operate in one of three modes: SIP trunk in IPv4-only mode, SIP trunk in IPv6-only mode, and SIP trunk in dual-stack mode, which supports both IPv4 and IPv6.

A SIP trunk uses the Alternative Network Address Transport (ANAT) mechanism to exchange multiple IPv4 and IPv6 media addresses for the endpoints in a session. ANAT is automatically enabled on SIP trunks in dual-stack mode. The ANAT Session Description Protocol (SDP) grouping framework allows user agents (UAs) to include both IPv4 and IPv6 addresses in their SDP session descriptions. The UA is then able to use any of its media addresses to establish a media session with a remote UA.

A Cisco Unified Border Element can interoperate between H.323/SIP IPv4 and SIP IPv6 networks in media flow-through mode. In media flow-through mode, both signaling and media flows through the Cisco Unified Border Element, and the Cisco Unified Border Element performs both signaling and media interoperation between H.323/SIP IPv4 and SIP IPv6 networks (see [Figure 1](#)).

Figure 1 *H.323/SIP IPv4—SIP IPv6 Interoperating in Media Flow-Through Mode*



The tasks in the following sections describe how to configure a SIP voice gateway for IPv6:

- [Shutting Down or Enabling VoIPv6 Service on Cisco Gateways, page 4](#)
- [Shutting Down or Enabling VoIPv6 Submodes on Cisco Gateways, page 5](#)
- [Configuring the Protocol Mode of the SIP Stack, page 6](#)
- [Configuring the Source IPv6 Address of Signaling and Media Packets, page 8](#)
- [Configuring the SIP Server, page 9](#)
- [Configuring SIP Register Support, page 11](#)
- [Configuring Outbound Proxy Server Globally on a SIP Gateway, page 12](#)
- [Verifying SIP Gateway Status, page 13](#)
- [Configuring H.323 IPv4-to-SIPv6 Connections in a Cisco Unified Border Element, page 15](#)

Restrictions

Virtual routing and forwarding (VRF) is not supported in IPv6 calls.

Shutting Down or Enabling VoIPv6 Service on Cisco Gateways

The following task describes how to shut down or enable VoIPv6 service on Cisco gateways.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **shutdown [forced]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.
Step 4	shutdown [forced] Example: Router(config-voi-serv)# shutdown forced	Shuts down or enables VoIP call services.

Shutting Down or Enabling VoIPv6 Submodes on Cisco Gateways

The following task describes how to shut down or enable VoIPv6 submodes on Cisco gateways.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **call service stop [forced] [maintain-registration]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.
Step 4	sip Example: Router(config-voi-serv)# sip	Enters SIP configuration mode.
Step 5	call service stop [forced] [maintain-registration] Example: Router(config-serv-sip)# call service stop	Shuts down or enables VoIPv6 for the selected submode.

Configuring the Protocol Mode of the SIP Stack

This task explains how to configure the SIP stack's protocol mode.

Prerequisites

SIP service should be shut down before configuring the protocol mode. After configuring the protocol mode as IPv6, IPv4, or dual-stack, SIP service should be reenabled.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **protocol mode {ipv4 | ipv6 | dual-stack [preference {ipv4 | ipv6}]}**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	sip-ua Example: Router(config)# sip-ua	Enters SIP user agent configuration mode.
Step 4	protocol mode { ipv4 ipv6 dual-stack [preference { ipv4 ipv6 }] } Example: Router(config-sip-ua)# protocol mode dual-stack	Configures the Cisco IOS SIP stack in dual-stack mode.

Disabling ANAT Mode

ANAT is automatically enabled on SIP trunks in dual-stack mode. This task describes how to disable ANAT in order to use a single-stack mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **no anat**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.

	Command or Action	Purpose
Step 4	sip Example: Router(config-voi-serv)# sip	Enters SIP configuration mode.
Step 5	no anat Example: router(conf-serv-sip)# no anat	Disables ANAT on a SIP trunk.

Configuring the Source IPv6 Address of Signaling and Media Packets

Users can configure the source IPv4 or IPv6 address of signaling and media packets to a specific interface's IPv4 or IPv6 address. Thus, the address that goes out on the packet is bound to the IPv4 or IPv6 address of the interface specified with the **bind** command.

The **bind** command also can be configured with one IPv6 address to force the gateway to use the configured address when the bind interface has multiple IPv6 addresses. The bind interface should have both IPv4 and IPv6 addresses to send out ANAT.

When you do not specify a bind address or if the interface is down, the IP layer still provides the best local address.

This task describes how to configure the source IPv6 address of signaling and media packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **bind { control | media | all } source-interface interface-id [ipv6-address ipv6-address]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.

	Command or Action	Purpose
Step 4	sip Example: Router(config-voi-serv)# sip	Enters SIP configuration mode.
Step 5	bind {control media all} source-interface interface-id [ipv6-address ipv6-address] Example: Router(config-serv-sip)# bind control source-interface FastEthernet0/0	Binds the source address for signaling and media packets to the IPv6 address of a specific interface.

Configuring the SIP Server

This task describes how to configure a SIP server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **sip-server {dns:[host-name] | ipv4:ipv4-address | ipv6:[ipv6-address][:port-num]}**
5. **keepalive target { {ipv4:address | ipv6:address} | [:port] | dns:hostname} [tcp [tls]] | udp [secondary]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	sip-ua Example: Router(config)# sip-ua	Enters SIP user agent configuration mode.

	Command or Action	Purpose
Step 4	sip-server { dns : <i>[host-name]</i> ipv4 : <i>ipv4-address</i> ipv6 : <i>[ipv6-address]:[port-nums]</i> } Example: Router(config-sip-ua)# sip-server ipv6:[2001:0DB8:0:0:8:800:200C:417A]	Configures a network address for the SIP server interface.
Step 5	keepalive target {{ ipv4 : <i>address</i> ipv6 : <i>address</i>][: <i>port</i>] dns : <i>hostname</i> } [tcp tls] udp] [secondary] Example: Router(config-sip-ua)# keepalive target ipv6:[2001:0DB8:0:0:8:800:200C:417A]	Identifies SIP servers that will receive keepalive packets from the SIP gateway.

Configuring the Session Target

This task describes how to configure the session target.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **dial-peer voice** *tag* {**mmoip** | **pots** | **vofr** | **voip**}
4. **destination-pattern** [+]*string*[**T**]
5. **session target** {**ipv4**:*destination-address* | **ipv6**:*[destination-address]* | **dns**:*[\$\$.| \$d\$.| \$e\$.| \$u\$.| host-name* | **enum**:*table-num* | **loopback**:**rtp** | **ras** | **sip-server**} [:*port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	dial-peer voice <i>tag</i> { mmoip pots vofr voip } Example: Router(config)# dial-peer voice 29 voip	Defines a particular dial peer, specifies the method of voice encapsulation, and enters dial peer configuration mode.

	Command or Action	Purpose
Step 4	destination-pattern <i>[+]</i> <i>string</i> [T] Example: Router(config-dial-peer)# destination-pattern 7777	Specifies either the prefix or the full E.164 telephone number to be used for a dial peer.
Step 5	session target { ipv4: <i>destination-address</i> ipv6: <i>[destination-address]</i> dns: <i>[\$s\$. \$d\$. \$e\$. \$u\$.]</i> <i>host-name</i> enum: <i>table-num</i> loopback: <i>rtp</i> ras sip-server } <i>[:port]</i> Example: Router(config-dial-peer)# session target [ipv6:2001:0DB8:0:0:8:800:200C:417A]	Designates a network-specific address to receive calls from a VoIP or VoIPv6 dial peer.

Configuring SIP Register Support

This task describes how to configure SIP register support.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sip-ua**
4. **registrar** {**dns:***address* | **ipv4:***destination-address* *[:port]* | **ipv6:***destination-address* *[:port]*} **aor-domain** **expires** *seconds* [**tcp** [**tls**]] **type** [**secondary**] [**scheme** *string*]
5. **retry register** *retries*
6. **timers register** *milliseconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	sip-ua Example: Router(config)# sip-ua	Enters SIP user agent configuration mode.

	Command or Action	Purpose
Step 4	registrar { dns:address ipv4:destination-address [: port] ipv6:destination-address [: port]} aor-domain expires seconds [tcp [tls]] type [secondary] [scheme string] Example: Router(config-sip-ua)# registrar ipv6:[3FFE:501:FFFF:5:20F:F7FF:FE0B:2972] expires 3600 secondary	Enables SIP gateways to register E.164 numbers on behalf of analog telephone voice ports, IP phone virtual voice ports, and SCCP phones with an external SIP proxy or SIP registrar.
Step 5	retry register <i>retries</i> Example: Router(config-sip-ua)# retry register 10	Configures the total number of SIP register messages that the gateway should send.
Step 6	timers register <i>milliseconds</i> Example: Router(config-sip-ua)# timers register 500	Configures how long the SIP UA waits before sending register requests.

Configuring Outbound Proxy Server Globally on a SIP Gateway

This task describes how to configure an outbound-proxy server globally on a SIP gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **sip**
5. **outbound-proxy** {**ipv4:ipv4-address** | **ipv6:[ipv6-address]** | **dns:host:domain**} [:**port-number**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.

	Command or Action	Purpose
Step 4	sip Example: Router(config-voi-serv)# sip	Enters sip configuration mode.
Step 5	outbound-proxy { ipv4 : <i>ipv4-address</i> ipv6 : <i>ipv6-address</i> dns : <i>host:domain</i> } [: <i>port-number</i>] Example: Router(config-serv-sip)# outbound-proxy ipv6 [2001:0DB8:0:0:8:800:200C:417A]	Specifies the SIP outbound proxy globally for a Cisco IOS voice gateway using an IPv6 address.

Verifying SIP Gateway Status

To verify SIP gateway information, use the following optional commands as needed:

- [show sip-ua calls, page 13](#)
- [show sip-ua connections, page 14](#)
- [show sip-ua status, page 14](#)

show sip-ua calls

The **show sip-ua calls** command displays active user agent client (UAC) and user agent server (UAS) information on SIP calls:

```
Router# show sip-ua calls
```

```
SIP UAC CALL INFO
```

```
Call 1
```

```
SIP Call ID           : 8368ED08-1C2A11DD-80078908-BA2972D0@2001::21B:D4FF:FED7:B000
  State of the call    : STATE_ACTIVE (7)
  Substate of the call : SUBSTATE_NONE (0)
  Calling Number       : 2000
  Called Number        : 1000
  Bit Flags            : 0xC04018 0x100 0x0
  CC Call ID          : 2
  Source IP Address (Sig) : 2001::21B:D4FF:FED7:B000
  Destn SIP Req Addr:Port : [2001::21B:D5FF:FE1D:6C00]:5060
  Destn SIP Resp Addr:Port : [2001::21B:D5FF:FE1D:6C00]:5060
  Destination Name     : 2001::21B:D5FF:FE1D:6C00
  Number of Media Streams : 1
  Number of Active Streams: 1
  RTP Fork Object      : 0x0
  Media Mode           : flow-through
  Media Stream 1
    State of the stream : STREAM_ACTIVE
    Stream Call ID      : 2
    Stream Type         : voice-only (0)
    Stream Media Addr Type : 1709707780
    Negotiated Codec    : (20 bytes)
    Codec Payload Type  : 18
    Negotiated Dtmf-relay : inband-voice
    Dtmf-relay Payload Type : 0
    Media Source IP Addr:Port : [2001::21B:D4FF:FED7:B000]:16504
    Media Dest IP Addr:Port  : [2001::21B:D5FF:FE1D:6C00]:19548
```

```
Options-Ping      ENABLED:NO      ACTIVE:NO
  Number of SIP User Agent Client(UAC) calls: 1

SIP UAS CALL INFO

  Number of SIP User Agent Server(UAS) calls: 0
```

show sip-ua connections

Use the **show sip-ua connections** command to display SIP UA transport connection tables:

```
Router# show sip-ua connections udp brief
```

```
Total active connections      : 1
No. of send failures          : 0
No. of remote closures        : 0
No. of conn. failures         : 0
No. of inactive conn. ageouts : 0
```

```
Router# show sip-ua connections udp detail
```

```
Total active connections      : 1
No. of send failures          : 0
No. of remote closures        : 0
No. of conn. failures         : 0
No. of inactive conn. ageouts : 0
```

```
-----Printing Detailed Connection Report-----
```

Note:

```
** Tuples with no matching socket entry
  - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port>'
    to overcome this error condition
++ Tuples with mismatched address/port entry
  - Do 'clear sip <tcp[tls]/udp> conn t ipv4:<addr>:<port> id <connid>'
    to overcome this error condition
```

```
Remote-Agent:2001::21B:D5FF:FE1D:6C00, Connections-Count:1
```

```
Remote-Port Conn-Id Conn-State WriteQ-Size
=====
5060          2 Established          0
```

show sip-ua status

Use the **show sip-ua status** command to display the status of the SIP UA:

```
Router# show sip-ua status
```

```
SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED

SIP User Agent for TLS over TCP : ENABLED
SIP User Agent bind status(signaling): DISABLED
SIP User Agent bind status(media): DISABLED
SIP early-media for 180 responses with SDP: ENABLED
SIP max-forwards : 70
SIP DNS SRV version: 2 (rfc 2782)
NAT Settings for the SIP-UA
Role in SDP: NONE
Check media source packets: DISABLED
Maximum duration for a telephone-event in NOTIFYs: 2000 ms
SIP support for ISDN SUSPEND/RESUME: ENABLED
Redirection (3xx) message handling: ENABLED
```

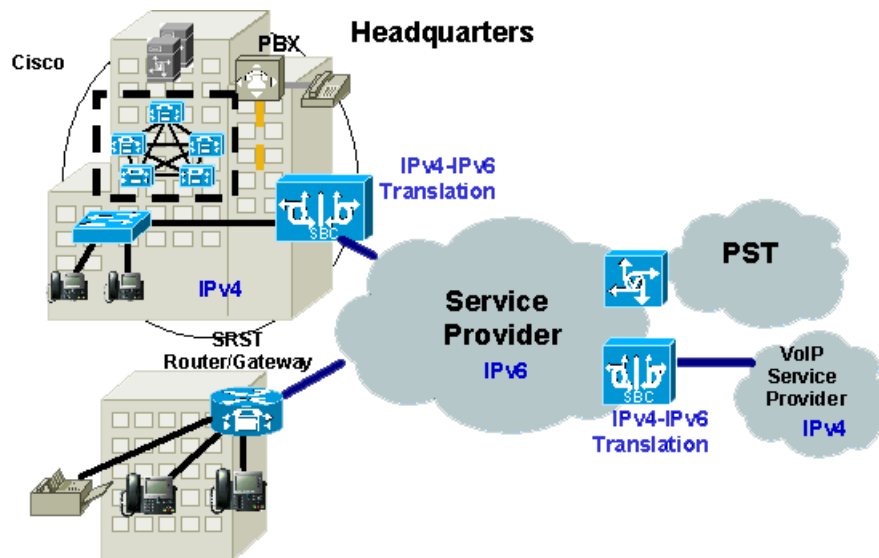
```
Reason Header will override Response/Request Codes: DISABLED
Out-of-dialog Refer: DISABLED
Presence support is DISABLED
protocol mode is ipv6
```

```
SDP application configuration:
Version line (v=) required
Owner line (o=) required
Timespec line (t=) required
Media supported: audio video image
Network types supported: IN
Address types supported: IP4 IP6
Transport types supported: RTP/AVP udptl
```

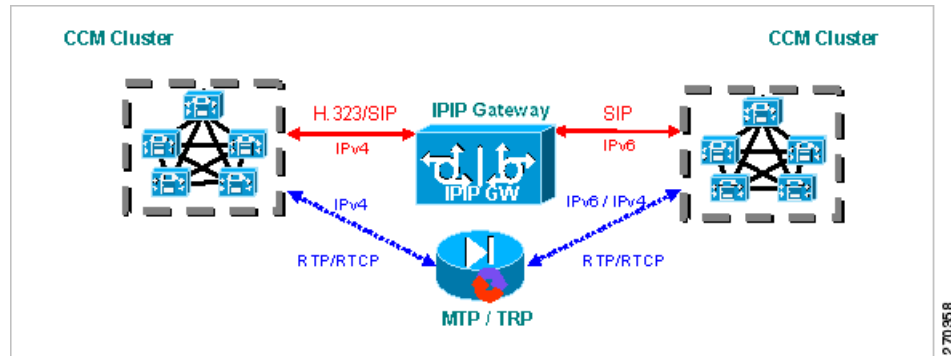
Configuring H.323 IPv4-to-SIPv6 Connections in a Cisco Unified Border Element

An organization with an IPv4 network can deploy a Cisco Unified Border Element on the boundary to connect with the service provider's IPv6 network (see [Figure 2](#)).

Figure 2 *Cisco Unified Border Element Interoperating IPv4 Networks with IPv6 Service Provider*



A Cisco Unified Border Element can interoperate between H.323/SIP IPv4 and SIP IPv6 networks in media flow-through mode. In media flow-through mode, both signaling and media flows through the Cisco Unified Border Element, and the Cisco Unified Border Element performs both signaling and media interoperation between H.323/SIP IPv4 and SIP IPv6 networks (see [Figure 3](#)).

Figure 3 IPv4 to IPv6 Media Interoperating Through Cisco IOS MTP

The Cisco Unified Border Element feature adds IPv6 capability to existing VoIP features. This feature adds dual-stack support on voice gateways and MTP, IPv6 support for SIP trunks, and SCCP-controlled analog voice gateways. In addition, the SBC functionality of connecting SIP IPv4 or H.323 IPv4 network to a SIP IPv6 network is implemented on an Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6.

To configure H.323 IPv4-to-SIPv6 connections in an Cisco Unified Border Element, perform the following task:

Prerequisites

Cisco Unified Border Element must be configured in IPv6-only or dual-stack mode to support IPv6 calls.

Restrictions

A Cisco Unified Border Element interoperates between H.323/SIP IPv4 and SIP IPv6 networks only in media flow-through mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **voice service voip**
4. **allow-connections** *from-type* **to** *to-type*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.
	Example: Router> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal	Enters global configuration mode.
	Example: Router# configure terminal	

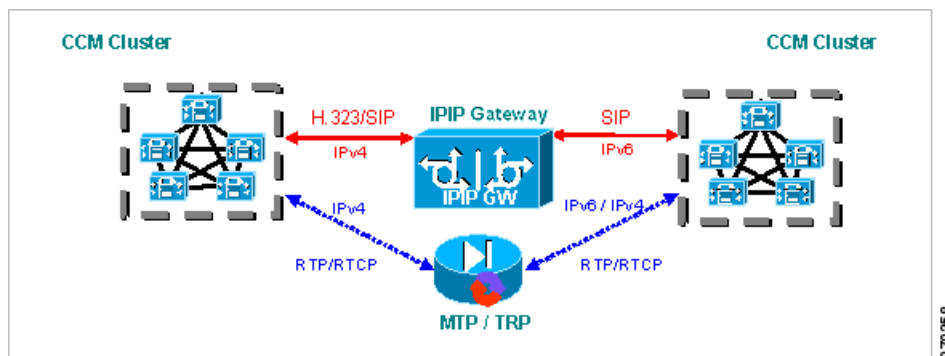
	Command or Action	Purpose
Step 3	voice service voip Example: Router(config)# voice service voip	Enters voice service VoIP configuration mode.
Step 4	allow-connections from-type to to-type Example: Router(config-voi-serv)# allow-connections h323 to sip	Allows connections between specific types of endpoints in a VoIPv6 network. Arguments are as follows: <ul style="list-style-type: none"> <i>from-type</i>—Type of connection. Valid values: h323, sip. <i>to-type</i>—Type of connection. Valid values: h323, sip.

Configuring MTP Used with Voice Gateways

Cisco IOS MTP trusted relay point (TRP) supports media interoperation between IPv4 and IPv6 networks (see Figure 4). This functionality is used when an IPv4 phone (registered to Cisco Unified Communications Manager, formerly known as Cisco Unified Call Manager) communicates with an IPv6 phone (registered to another Cisco Unified Communications Manager). In this case, one of the Cisco Unified Communications Managers inserts a Cisco IOS MTP to perform the IPv4-to-IPv6 media translation between the phones.

MTP for IPv4-to-IPv6 media translation operates only in dual-stack mode. Communication between Cisco IOS MTP and Cisco Unified Communications Manager occurs over SCCP for IPv4 only.

Figure 4 IPv4 to IPv6 Media Interoperating Through Cisco IOS MTP



The VoIPv6 feature includes IPv4 and IPv6 dual-stack support on voice gateways and MTP, IPv6 support for SIP trunks, and SCCP-controlled analog phones. In addition, connecting a SIP IPv4 or H.323 IPv4 network to a SIP IPv6 network is implemented on Cisco Unified Border Element.

To configure IPv6 with media interoperating using Cisco Unified Communications Manager-controlled MTP, perform the following task in the following sections:

- [Configuring MTP for IPv4-to-IPv6 Translation, page 18](#)

Restrictions

- MTP for IPv4-to-IPv6 media translation operates in dual-stack mode only.

- A SIP trunk can be configured over IPv4 only, over IPv6 only, or in dual-stack mode. In dual-stack mode, ANAT is used to describe both IPv4 and IPv6 media capabilities.

Configuring MTP for IPv4-to-IPv6 Translation

This task describes how to configure MTP for IPv4-to-IPv6 translation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **sccp ccm** {*ipv4-address* | *ipv6-address* | *dns*} **identifier** *identifier-number* [**priority** *priority*] [**port** *port-number*] [**version** *version-number*]
4. **sccp ccm group** *group-number*
5. **associate profile** *profile-identifier* **register** *device-name*
6. **exit**
7. **dspfarm profile** *profile-identifier* {**conference** | **mtp** | **transcode**} [**security**]
8. **codec** {*codec-type* | **pass-through**}
9. **maximum sessions** {**hardware** | **software**} *number*
10. **associate application** **sccp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	sccp ccm { <i>ipv4-address</i> <i>ipv6-address</i> <i>dns</i> } identifier <i>identifier-number</i> [priority <i>priority</i>] [port <i>port-number</i>] [version <i>version-number</i>] Example: Router(global)# sccp ccm 2001:DB8:C18:1::102 identifier 2 version 7.0	Adds a Cisco Unified CallManager server to the list of available servers and set various parameters—including IP address, IPv6 address, or Domain Name System (DNS) name, port number, and version number. Note SCCP communication between Cisco IOS MTP and Cisco Unified Border Element is supported only for an IPv4-only network. Do not use the <i>ipv6-address</i> argument with this command if you are configuring for the Cisco Unified Border Element.
Step 4	sccp ccm group <i>group-number</i> Example: Router(global)# sccp ccm group 1	Creates a Cisco CallManager group and enters SCCP Cisco CallManager configuration mode

	Command or Action	Purpose
Step 5	associate profile <i>profile-identifier</i> register <i>device-name</i> Example: Router(config-sccp-ccm)# associate profile 5 register MTP3825	Associates a digital signal processor (DSP) farm profile with a Cisco CallManager group.
Step 6	exit Example: Router(config-sip-ua)# exit	Exits the current configuration mode.
Step 7	dspfarm profile <i>profile-identifier</i> { conference mtp transcode } [security] Example: Router(config)# dspfarm profile 5 mtp	Enters DSP farm profile configuration mode and defines a profile for DSP farm services.
Step 8	codec { <i>codec-type</i> pass-through } Example: Router(config-dspfarm-profile)# codec g711ulaw	Specifies the codecs that are supported by a DSP farm profile.
Step 9	maximum sessions { hardware software } <i>number</i> Example: Router(config-dspfarm-profile)# maximum sessions software 100	Specifies the maximum number of sessions that are supported by the profile.
Step 10	associate application sccp Example: Router(config-dspfarm-profile)# associate application SCCP	Associates SCCP to the DSP farm profile.

Configuration Examples for Implementing VoIP over IPv6

This section contains the following configuration examples for the VoIP over IPv6 feature.

- [Configuring the SIP Trunk: Example, page 20](#)
- [Configuring the Source IPv6 Address of Signaling and Media Packets: Example, page 20](#)
- [Configuring the SIP Server: Example, page 20](#)
- [Configuring the Session Target: Example, page 20](#)
- [Configuring SIP Register Support: Example, page 20](#)
- [Configuring H.323 IPv4 to SIPv6 Connections in a Cisco Unified Border Element: Example, page 20](#)
- [Configuring MTP for IPv4-to-IPv6 Translation: Example, page 21](#)

Configuring the SIP Trunk: Example

This example shows how to configure the SIP trunk to use dual-stack mode, with IPv6 as the preferred mode. The SIP service must be shut down before any changes are made to protocol mode configuration.

```
Router(config)# sip-ua
Router(config-sip-ua)# protocol mode dual-stack preference ipv6
```

Configuring the Source IPv6 Address of Signaling and Media Packets: Example

This example shows how to configure the **bind** command:

```
Router(config)# voice service voip
Router(config-voi-serv)# sip
Router(config-serv-sip)# bind control source-interface FastEthernet 0/0
```

Configuring the SIP Server: Example

This example shows how to configure the SIP server:

```
Router(config)# sip-ua
Router(config-sip-ua)# sip-server ipv6:[2001:0DB8:0:0:8:800:200C:417A]
```

Configuring the Session Target: Example

This example shows how to configure the session target:

```
Router(config)# dial-peer voice 29 voip
Router(config-dial-peer)# destination-pattern 7777
Router(config-dial-peer)# session target ipv6:[2001:0DB8:0:0:8:800:200C:417A]
```

Configuring SIP Register Support: Example

This example shows how to configure SIP register support:

```
Router(config)# sip-ua
Router(config-sip-ua)# registrar ipv6:[2001:0DB8:0:0:8:800:200C:417A] expires 3600
secondary
Router(config-sip-ua)# retry register 10
Router(config-sip-ua)# timers register 500
```

Configuring H.323 IPv4 to SIPv6 Connections in a Cisco Unified Border Element: Example

This example shows how to configure H.323 IPv4 to IPv6 connections in a Cisco Unified Border Element.

```
Router(config)# voice service voip
Router(config-voi-serv)# allow-connections h323 to sip
```

Configuring MTP for IPv4-to-IPv6 Translation: Example

The following example shows how to configure MTP for IPv4-to-IPv6 translation and provides sample configuration output:

```
Router(global)# sccp ccm group 1
Router(config-sccp-ccm)# associate profile 5 register MTP3825
Router(config-sccp-ccm)# exit
Router(config)# dspfarm profile 5 mtp
Router(config-dspfarm-profile)# codec g711ulaw
Router(config-dspfarm-profile)# maximum sessions software 100
Router(config-dspfarm-profile)# associate application SCCP
```

```
Router# show sccp
```

```
sccp ccm group 1
associate profile 5 register MTP3825
!
dspfarm profile 5 mtp
  codec g711ulaw
  maximum sessions software 100
  associate application SCCP
```

Additional References

The following sections provide references related to the Implementing VoIP for IPv6 feature.

Related Documents

Related Topic	Document Title
Cisco Express Forwarding for IPv6	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPv4-to-IPv6 media translation	“Configuring Cisco IOS Hosted NAT Traversal for Session Border Controller,” Cisco IOS NAT Configuration Guide
Cisco IOS voice configuration	Cisco IOS Voice Configuration Library
Cisco Unified Border Element configuration	Cisco Unified Border Element Configuration Guide
Cisco Unified Communications Manager	Cisco Unified Communications Manager
Dual-stack information and configuration	“Implementing IPv6 Addressing and Basic Connectivity,” Cisco IOS IPv6 Configuration Guide
IPv4 VoIP gateway	VoIP Gateway Trunk and Carrier Based Routing Enhancements
VoIPv4 dial peer information and configuration	Dial Peer Features and Configuration
SIP information	SIP Configuration Guide
SIP bind information	Configuring SIP Bind Features
Basic H.323 gateway configuration	“Configuring H.323 Gateways,” Cisco IOS Voice, Video, and Fax Configuration Guide
Basic H.323 gatekeeper configuration	“Configuring H.323 Gatekeepers,” Cisco IOS Voice, Video, and Fax Configuration Guide
IPv6 commands, including voice commands	Cisco IOS IPv6 Command Reference
Troubleshooting and debugging guides	<ul style="list-style-type: none"> • Cisco IOS Debug Command Reference • Troubleshooting and Debugging VoIP Call Basics • VoIP Debug Commands

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 3095	<i>RObust Header Compression (ROHC): Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed</i>
RFC 3759	<i>RObust Header Compression (ROHC): Terminology and Channel Mapping Examples</i>
RFC 4091	<i>The Alternative Network Address Types (ANAT) Semantics for the Session Description Protocol (SDP) Grouping Framework</i>
RFC 4092	<i>Usage of the Session Description Protocol (SDP) Alternative Network Address Types (ANAT) Semantics in the Session Initiation Protocol (SIP)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/all_release/all_mcl.html.

- **allow-connections**
- **anat**
- **associate application sccp**
- **associate profile**
- **bind**
- **call service stop**
- **codec (DSP farm profile)**
- **destination-pattern**
- **dial-peer voice**
- **dspfarm profile**
- **ip source-address (telephony-service)**
- **keepalive target**
- **maximum sessions (DSP farm profile)**
- **outbound-proxy**
- **port (dial-peer)**
- **protocol mode**
- **registrar**
- **retry register**
- **sccp ccm**
- **session target (VoIP dial peer)**
- **show call active fax**
- **show call active voice**
- **show call history fax**
- **show call history voice**
- **show dial-peer voice**
- **show sccp**
- **show_sip-ua_calls**
- **show sip-ua connections**
- **show sip-ua status**
- **show stcapp device**
- **show voip rtp connections**
- **shutdown (voice-port)**
- **sip-server**
- **timers register**
- **translation-profile (dial-peer)**
- **voice-class sip anat**
- **voice-class source interface**

Feature Information for Implementing VoIP for IPv6

Table 1 lists the release history for this feature.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Implementing VoIP for IPv6

Feature Name	Releases	Feature Information
VoIP for IPv6	12.4(22)T	<p>VoIPv6 adds IPv6 capability to existing VoIP features. VoIPv6 requires IPv6 and IPv4 dual-stack support on voice gateways and MTP, IPv6 support for SIP trunks, and SCCP-controlled analog voice phones. In addition, the SBC functionality of connecting SIP IPv4 or H.323 IPv4 network to SIP IPv6 network is implemented on a Cisco Unified Border Element to facilitate migration from VoIPv4 to VoIPv6.</p> <p>This entire document provides information about this feature.</p>

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2008–2009 Cisco Systems, Inc. All rights reserved.



Implementing Tunneling for IPv6

First Published: June 7, 2001

Last Updated: August 18, 2008

This module describes how to configure overlay tunneling techniques used by the Cisco IOS software to support the transition from IPv4-only networks to integrated IPv4- and IPv6-based networks. Tunneling encapsulates IPv6 packets in IPv4 packets and uses the IPv4 network as a link-layer mechanism.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the [“Feature Information for Implementing Tunneling for IPv6” section on page 24](#).

Use Cisco Feature Navigator to find information about platform support and Cisco IOS, Catalyst OS, and Cisco IOS XE software image support. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.

Contents

- [Restrictions for Implementing Tunneling for IPv6, page 2](#)
- [Information About Implementing Tunneling for IPv6, page 2](#)
- [How to Implement Tunneling for IPv6, page 7](#)
- [Configuration Examples for Implementing Tunneling for IPv6, page 17](#)
- [Where to Go Next, page 21](#)
- [Additional References, page 21](#)
- [Command Reference, page 23](#)
- [Feature Information for Implementing Tunneling for IPv6, page 24](#)



Americas Headquarters:

Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

© 2006–2009 Cisco Systems, Inc. All rights reserved.

Restrictions for Implementing Tunneling for IPv6

- In Cisco IOS Release 12.0(21)ST and Cisco IOS Release 12.0(22)S and earlier releases, the Cisco 12000 series gives a very low priority to the processing of IPv6 tunneled packets. Therefore, we strongly recommend that you limit the use of IPv6 tunnels on the Cisco 12000 series using these releases to topologies that sustain a low level of network traffic and require a minimal amount of process-switching resources.
- IPv6 manually configured tunnel traffic in Cisco IOS Release 12.0(23)S is processed in software on the CPU of the line card, instead of in the Route Processor (RP) in the Cisco 12000 router, resulting in enhanced performance.

Information About Implementing Tunneling for IPv6

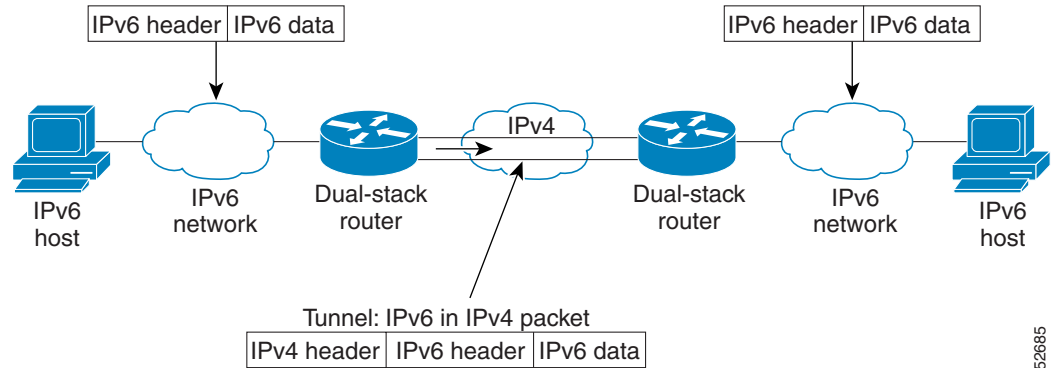
To configure tunneling for IPv6, you need to understand the following concepts:

- [Overlay Tunnels for IPv6, page 2](#)
- [IPv6 Manually Configured Tunnels, page 4](#)
- [GRE/IPv4 Tunnel Support for IPv6 Traffic, page 4](#)
- [GRE/CLNS Tunnel Support for IPv4 and IPv6 Packets, page 5](#)
- [Automatic 6to4 Tunnels, page 5](#)
- [Automatic IPv4-Compatible IPv6 Tunnels, page 5](#)
- [ISATAP Tunnels, page 6](#)
- [IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface, page 6](#)

Overlay Tunnels for IPv6

Overlay tunneling encapsulates IPv6 packets in IPv4 packets for delivery across an IPv4 infrastructure (a core network or the Internet) (see [Figure 1](#)). By using overlay tunnels, you can communicate with isolated IPv6 networks without upgrading the IPv4 infrastructure between them. Overlay tunnels can be configured between border routers or between a border router and a host; however, both tunnel endpoints must support both the IPv4 and IPv6 protocol stacks. Cisco IOS IPv6 supports the following types of overlay tunneling mechanisms:

- Manual
- Generic routing encapsulation (GRE)
- IPv4-compatible
- 6to4
- Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)

Figure 1 **Overlay Tunnels****Note**

Overlay tunnels reduce the maximum transmission unit (MTU) of an interface by 20 octets (assuming the basic IPv4 packet header does not contain optional fields). A network using overlay tunnels is difficult to troubleshoot. Therefore, overlay tunnels connecting isolated IPv6 networks should not be considered as a final IPv6 network architecture. The use of overlay tunnels should be considered as a transition technique toward a network that supports both the IPv4 and IPv6 protocol stacks or just the IPv6 protocol stack.

Use [Table 1](#) to help you determine which type of tunnel you want to configure to carry IPv6 packets over an IPv4 network.

Table 1 **Suggested Usage of Tunnel Types to Carry IPv6 Packets over an IPv4 Network**

Tunneling Type	Suggested Usage	Usage Notes
Manual	Simple point-to-point tunnels that can be used within a site or between sites	Can carry IPv6 packets only.
GRE- and IPv4-compatible	Simple point-to-point tunnels that can be used within a site or between sites	Can carry IPv6, Connectionless Network Service (CLNS), and many other types of packets.
IPv4-compatible	Point-to-multipoint tunnels	Uses the ::/96 prefix. We do not now recommend using this tunnel type.
6to4	Point-to-multipoint tunnels that can be used to connect isolated IPv6 sites	Sites use addresses from the 2002::/16 prefix.
ISATAP	Point-to-multipoint tunnels that can be used to connect systems within a site	Sites can use any IPv6 unicast addresses.

Individual tunnel types are discussed in detail in this document. We recommend that you review and understand the information about the specific tunnel type that you want to implement. When you are familiar with the type of tunnel you need, see [Table 2](#) for a summary of the tunnel configuration parameters that you may find useful.

Table 2 *Tunnel Configuration Parameters by Tunneling Type*

Tunneling Type	Tunnel Configuration Parameter			
	Tunnel Mode	Tunnel Source	Tunnel Destination	Interface Prefix or Address
Manual	ipv6ip	An IPv4 address, or a reference to an interface on which IPv4 is configured.	An IPv4 address.	An IPv6 address.
GRE/IPv4	gre ip		An IPv4 address.	An IPv6 address.
IPv4-compatible	ipv6ip auto-tunnel		Not required. These are all point-to-multipoint tunneling types.	Not required. The interface address is generated as <code>::tunnel-source/96</code> .
6to4	ipv6ip 6to4		The IPv4 destination address is calculated, on a per-packet basis, from the IPv6 destination.	An IPv6 address. The prefix must embed the tunnel source IPv4 address
ISATAP	ipv6ip isatap			An IPv6 prefix in modified eui-64 format. The IPv6 address is generated from the prefix and the tunnel source IPv4 address.

IPv6 Manually Configured Tunnels

A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The primary use is for stable connections that require regular secure communication between two edge routers or between an end system and an edge router, or for connection to remote IPv6 networks.

An IPv6 address is manually configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks. Manually configured tunnels can be configured between border routers or between a border router and a host. Cisco Express Forwarding switching can be used for IPv6 manually configured tunnels, or Cisco Express Forwarding switching can be disabled if process switching is needed.

GRE/IPv4 Tunnel Support for IPv6 Traffic

IPv6 traffic can be carried over IPv4 GRE tunnels using the standard GRE tunneling technique that is designed to provide the services necessary to implement any standard point-to-point encapsulation scheme. As in IPv6 manually configured tunnels, GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol.

The primary use of GRE tunnels is for stable connections that require regular secure communication between two edge routers or between an edge router and an end system. The edge routers and the end systems must be dual-stack implementations.

GRE has a protocol field that identifies the passenger protocol. GRE tunnels allow Intermediate System-to-Intermediate System (IS-IS) or IPv6 to be specified as a passenger protocol, which allows both IS-IS and IPv6 traffic to run over the same tunnel. If GRE did not have a protocol field, it would be impossible to distinguish whether the tunnel was carrying IS-IS or IPv6 packets. The GRE protocol field is why it is desirable that you tunnel IS-IS and IPv6 inside GRE.

GRE/CLNS Tunnel Support for IPv4 and IPv6 Packets

GRE tunneling of IPv4 and IPv6 packets through CLNS networks enables Cisco CLNS Tunnels (CTunnels) to interoperate with networking equipment from other vendors. This feature provides compliance with RFC 3147.

The optional GRE services defined in header fields, such as checksums, keys, and sequencing, are not supported. Any packet received requesting such services will be dropped.

Refer to [Cisco IOS ISO CLNS Configuration Guide](#) for further information about this feature.

Automatic 6to4 Tunnels

An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The key difference between automatic 6to4 tunnels and manually configured tunnels is that the tunnel is not point-to-point; it is point-to-multipoint. In automatic 6to4 tunnels, routers are not configured in pairs because they treat the IPv4 infrastructure as a virtual nonbroadcast multiaccess (NBMA) link. The IPv4 address embedded in the IPv6 address is used to find the other end of the automatic tunnel.

An automatic 6to4 tunnel may be configured on a border router in an isolated IPv6 network, which creates a tunnel on a per-packet basis to a border router in another IPv6 network over an IPv4 infrastructure. The tunnel destination is determined by the IPv4 address of the border router extracted from the IPv6 address that starts with the prefix 2002::/16, where the format is `2002:border-router-IPv4-address::/48`. Following the embedded IPv4 address are 16 bits that can be used to number networks within the site. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks. 6to4 tunnels are configured between border routers or between a border router and a host.

The simplest deployment scenario for 6to4 tunnels is to interconnect multiple IPv6 sites, each of which has at least one connection to a shared IPv4 network. This IPv4 network could be the global Internet or a corporate backbone. The key requirement is that each site have a globally unique IPv4 address; the Cisco IOS software uses this address to construct a globally unique 6to4/48 IPv6 prefix. As with other tunnel mechanisms, appropriate entries in a Domain Name System (DNS) that map between hostnames and IP addresses for both IPv4 and IPv6 allow the applications to choose the required address.

Automatic IPv4-Compatible IPv6 Tunnels

Automatic IPv4-compatible tunnels use IPv4-compatible IPv6 addresses. IPv4-compatible IPv6 addresses are IPv6 unicast addresses that have zeros in the high-order 96 bits of the address, and an IPv4 address in the low-order 32 bits. They can be written as `0:0:0:0:0:A.B.C.D` or `::A.B.C.D`, where “A.B.C.D” represents the embedded IPv4 address.

The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an IPv4-compatible tunnel must support both the IPv4 and IPv6 protocol stacks. IPv4-compatible tunnels can be configured between border-routers or between a border-router and a host. Using IPv4-compatible tunnels is an easy method to create tunnels for IPv6 over IPv4, but the technique does not scale for large networks.

ISATAP Tunnels

ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6. ISATAP is designed for transporting IPv6 packets within a site where a native IPv6 infrastructure is not yet available; for example, when sparse IPv6 hosts are deployed for testing. ISATAP tunnels allow individual IPv4 or IPv6 dual-stack hosts within a site to communicate with other such hosts on the same virtual link, basically creating an IPv6 network using the IPv4 infrastructure.

The ISATAP router provides standard router advertisement network configuration support for the ISATAP site. This feature allows clients to automatically configure themselves as they would do if they were connected to an Ethernet. It can also be configured to provide connectivity out of the site. ISATAP uses a well-defined IPv6 address format composed of any unicast IPv6 prefix (/64), which can be link local, or global (including 6to4 prefixes), enabling IPv6 routing locally or on the Internet. The IPv4 address is encoded in the last 32 bits of the IPv6 address, enabling automatic IPv6-in-IPv4 tunneling.

Although the ISATAP tunneling mechanism is similar to other automatic tunneling mechanisms, such as IPv6 6to4 tunneling, ISATAP is designed for transporting IPv6 packets within a site, not between sites.

ISATAP uses unicast addresses that include a 64-bit IPv6 prefix and a 64-bit interface identifier. The interface identifier is created in modified EUI-64 format in which the first 32 bits contain the value 000:5EFE to indicate that the address is an IPv6 ISATAP address. [Table 3](#) describes an ISATAP address format.

Table 3 *IPv6 ISATAP Address Format*

64 Bits	32 Bits	32 Bits
link local or global IPv6 unicast prefix	0000:5EFE	IPv4 address of the ISATAP link

As shown in [Table 3](#), an ISATAP address consists of an IPv6 prefix and the ISATAP interface identifier. This interface identifier includes the IPv4 address of the underlying IPv4 link. The following example shows what an actual ISATAP address would look like if the prefix is 2001:0DB8:1234:5678::/64 and the embedded IPv4 address is 10.173.129.8. In the ISATAP address, the IPv4 address is expressed in hexadecimal as 0AAD:8108. For example, 2001:0DB8:1234:5678:0000:5EFE:0AAD:8108.

IPv6 IPsec Site-to-Site Protection Using Virtual Tunnel Interface

The IPv6 IPsec feature provides IPv6 crypto site-to-site protection of all types of IPv6 unicast and multicast traffic using native IPsec IPv6 encapsulation. The IPsec virtual tunnel interface (VTI) feature provides this function, using IKE as the management protocol.

An IPsec VTI supports native IPsec tunneling and includes most of the properties of a physical interface. The IPsec VTI alleviates the need to apply crypto maps to multiple interfaces and provides a routable interface.

The IPsec VTI allows IPv6 routers to work as security gateways, establish IPsec tunnels between other security gateway routers, and provide crypto IPsec protection for traffic from internal network when being transmitting across the public IPv6 Internet.

For further information on VTIs, see [Implementing IPsec in IPv6 Security](#).

How to Implement Tunneling for IPv6

The following sections explain how to implement tunneling for IPv6:

- [Configuring Manual IPv6 Tunnels, page 7](#)
- [Configuring GRE IPv6 Tunnels, page 8](#)
- [Configuring Automatic 6to4 Tunnels, page 10](#)
- [Automatic IPv4-Compatible IPv6 Tunnels, page 5](#)
- [Configuring ISATAP Tunnels, page 13](#)
- [Verifying IPv6 Tunnel Configuration and Operation, page 14](#)

Configuring Manual IPv6 Tunnels

This task explains how to configure a IPv6 overlay tunnel manually.

Prerequisites

With manually configured IPv6 tunnels, an IPv6 address is configured on a tunnel interface, and manually configured IPv4 addresses are assigned to the tunnel source and the tunnel destination. The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]
5. **tunnel source** { *ip-address* | *interface-type interface-number* }
6. **tunnel destination** *ip-address*
7. **tunnel mode ipv6ip**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>tunnel-number</i> Example: Router(config)# interface tunnel 0	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	ipv6 address <i>ipv6-prefix/prefix-length</i> [eiui-64] Example: Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	tunnel source { <i>ip-address</i> <i>interface-type interface-number</i> } Example: Router(config-if)# tunnel source ethernet 0	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. <ul style="list-style-type: none">If an interface is specified, the interface must be configured with an IPv4 address.
Step 6	tunnel destination <i>ip-address</i> Example: Router(config-if)# tunnel destination 192.168.30.1	Specifies the destination IPv4 address or hostname for the tunnel interface.
Step 7	tunnel mode ipv6ip Example: Router(config-if)# tunnel mode ipv6ip	Specifies a manual IPv6 tunnel. Note The tunnel mode ipv6ip command specifies IPv6 as the passenger protocol and IPv4 as both the encapsulation and transport protocol for the manual IPv6 tunnel.

Configuring GRE IPv6 Tunnels

This task explains how to configure a GRE tunnel on an IPv6 network. GRE tunnels can be configured to run over an IPv6 network layer and to transport IPv6 packets in IPv6 tunnels and IPv4 packets in IPv6 tunnels.

Prerequisites

When GRE IPv6 tunnels are configured, IPv6 addresses are assigned to the tunnel source and the tunnel destination. The tunnel interface can have either IPv4 or IPv6 addresses assigned (this is not shown in the task). The host or router at each end of a configured tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *ipv6-address* | *interface-type interface-number*}
6. **tunnel destination** {*host-name* | *ip-address* | *ipv6-address*}
7. **tunnel mode** {**aurp** | **cayman** | **dvmrp** | **eon** | **gre** | **gre multipoint** | **gre ipv6** | **ipip** | **decapsulate-any** | **iptalk** | **ipv6** | **mpls** | **nos**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>tunnel-number</i> Example: Router(config)# interface tunnel 0	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	ipv6 address <i>ipv6-prefix/prefix-length</i> [eui-64] Example: Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.
Step 5	tunnel source { <i>ip-address</i> <i>ipv6-address</i> <i>interface-type interface-number</i> } Example: Router(config-if)# tunnel source ethernet 0	Specifies the source IPv4 address or the source interface type and number for the tunnel interface. <ul style="list-style-type: none"> • If an interface is specified, the interface must be configured with an IPv4 address.

	Command or Action	Purpose
Step 6	tunnel destination { <i>host-name</i> <i>ip-address</i> <i>ipv6-address</i> } Example: Router(config-if)# tunnel destination 192.168.30.1	Specifies the destination IPv4 address or hostname for the tunnel interface.
Step 7	tunnel mode { <i>aurp</i> <i>cayman</i> <i>dvmrp</i> <i>eon</i> <i>gre</i> <i>gre multipoint</i> <i>gre ipv6</i> <i>ipip</i> <i>[decapsulate-any]</i> <i>iptalk</i> <i>ipv6</i> <i>mpls</i> <i>nos</i> } Example: Router(config-if)# tunnel mode gre ipv6	Specifies a GRE IPv6 tunnel. Note The tunnel mode gre ipv6 command specifies GRE as the encapsulation protocol for the tunnel.

Configuring Automatic 6to4 Tunnels

This task explains how to configure a 6to4 overlay tunnel.

Prerequisites

With 6to4 tunnels, the tunnel destination is determined by the border router IPv4 address, which is concatenated to the prefix 2002::/16 in the format 2002:*border-router-IPv4-address*::/48. The border router at each end of a 6to4 tunnel must support both the IPv4 and IPv6 protocol stacks.

Restrictions

The configuration of only one IPv4-compatible tunnel and one 6to4 IPv6 tunnel is supported on a router. If you choose to configure both of those tunnel types on the same router, we strongly recommend that they do not share the same tunnel source.

The reason that a 6to4 tunnel and an IPv4-compatible tunnel cannot share an interface is that both of them are NBMA “point-to-multipoint” access links and only the tunnel source can be used to reorder the packets from a multiplexed packet stream into a single packet stream for an incoming interface. So when a packet with an IPv4 protocol type of 41 arrives on an interface, that packet is mapped to an IPv6 tunnel interface based on the IPv4 address. However, if both the 6to4 tunnel and the IPv4-compatible tunnel share the same source interface, the router is not able to determine the IPv6 tunnel interface to which it should assign the incoming packet.

IPv6 manually configured tunnels can share the same source interface because a manual tunnel is a “point-to-point” link, and both the IPv4 source and IPv4 destination of the tunnel are defined.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]
5. **tunnel source** {*ip-address* | *interface-type interface-number*}

6. **tunnel mode ipv6ip 6to4**
7. **exit**
8. **ipv6 route** *ipv6-prefix/prefix-length* **tunnel** *tunnel-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel <i>tunnel-number</i> Example: Router(config)# interface tunnel 0	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	ipv6 address <i>ipv6-prefix/prefix-length</i> [ui-64] Example: Router(config-if)# ipv6 address 2002:c0a8:6301:1::1/64	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. <ul style="list-style-type: none"> The 32 bits following the initial 2002::/16 prefix correspond to an IPv4 address assigned to the tunnel source.
Step 5	tunnel source { <i>ip-address</i> <i>interface-type</i> <i>interface-number</i> } Example: Router(config-if)# tunnel source ethernet 0	Specifies the source interface type and number for the tunnel interface. Note The interface type and number specified in the tunnel source command must be configured with an IPv4 address.
Step 6	tunnel mode ipv6ip 6to4 Example: Router(config-if)# tunnel mode ipv6ip 6to4	Specifies an IPv6 overlay tunnel using a 6to4 address.

	Command or Action	Purpose
Step 7	exit Example: Router(config-if)# exit	Exits interface configuration mode, and returns the router to global configuration mode.
Step 8	ipv6 route <i>ipv6-prefix/prefix-length</i> tunnel <i>tunnel-number</i> Example: Router(config)# ipv6 route 2002::/16 tunnel 0	Configures a static route for the IPv6 6to4 prefix 2002::/16 to the specified tunnel interface. Note When configuring a 6to4 overlay tunnel, you must configure a static route for the IPv6 6to4 prefix 2002::/16 to the 6to4 tunnel interface. <ul style="list-style-type: none"> The tunnel number specified in the ipv6 route command must be the same tunnel number specified in the interface tunnel command.

Configuring IPv4-Compatible IPv6 Tunnels

This task explains how to configure an IPv4-compatible IPv6 overlay tunnel.

Prerequisites

With an IPv4-compatible tunnel, the tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of IPv4-compatible IPv6 addresses. The host or router at each end of an IPv4-compatible tunnel must support both the IPv4 and IPv6 protocol stacks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **tunnel source** {*ip-address* | *interface-type interface-number*}
5. **tunnel mode ipv6ip auto-tunnel**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	interface tunnel <i>tunnel-number</i> Example: Router(config)# interface tunnel 0	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	tunnel source { <i>ip-address</i> <i>interface-type interface-number</i> } Example: Router(config-if)# tunnel source ethernet 0	Specifies the source interface type and number for the tunnel interface. Note The interface type and number specified in the tunnel source command is configured with an IPv4 address only.
Step 5	tunnel mode ipv6ip auto-tunnel Example: Router(config-if)# tunnel mode ipv6ip auto-tunnel	Specifies an IPv4-compatible tunnel using an IPv4-compatible IPv6 address.

Configuring ISATAP Tunnels

This task describes how to configure an ISATAP overlay tunnel.

Prerequisites

The **tunnel source** command used in the configuration of an ISATAP tunnel must point to an interface with an IPv4 address configured. The ISATAP IPv6 address and prefix (or prefixes) advertised are configured as for a native IPv6 interface. The IPv6 tunnel interface must be configured with a modified EUI-64 address because the last 32 bits in the interface identifier are constructed using the IPv4 tunnel source address.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *tunnel-number*
4. **ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]
5. **no ipv6 nd suppress-ra**
6. **tunnel source** {*ip-address* | *interface-type interface-number*}
7. **tunnel mode ipv6ip isatap**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	interface tunnel tunnel-number Example: Router(config)# interface tunnel 1	Specifies a tunnel interface and number, and enters interface configuration mode.
Step 4	ipv6 address ipv6-prefix/prefix-length [eui-64] Example: Router(config-if)# ipv6 address 2001:0DB8:6301::/64 eui-64	Specifies the IPv6 address assigned to the interface and enables IPv6 processing on the interface. Note Refer to the <i>Configuring Basic Connectivity for IPv6</i> module for more information on configuring IPv6 addresses.
Step 5	no ipv6 nd suppress-ra Example: Router(config-if)# no ipv6 nd suppress-ra	Sending of IPv6 router advertisements is disabled by default on tunnel interfaces. This command reenables the sending of IPv6 router advertisements to allow client autoconfiguration.
Step 6	tunnel source {ip-address interface-type interface-number} Example: Router(config-if)# tunnel source ethernet 1/0/1	Specifies the source interface type and number for the tunnel interface. Note The interface type and number specified in the tunnel source command must be configured with an IPv4 address.
Step 7	tunnel mode ipv6ip isatap Example: Router(config-if)# tunnel mode ipv6ip isatap	Specifies an IPv6 overlay tunnel using a ISATAP address.

Verifying IPv6 Tunnel Configuration and Operation

This optional task explains how to verify IPv6 tunnel configuration and operation. The commands contained in the task steps can be used in any sequence and may need to be repeated.

SUMMARY STEPS

1. **enable**
2. **show interfaces tunnel number [accounting]**
3. **ping [protocol] destination**

4. `show ip route` [*address* [*mask*]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	show interfaces tunnel <i>number</i> [accounting] Example: Router# show interfaces tunnel 0	(Optional) Displays tunnel interface information. <ul style="list-style-type: none"> Use the <i>number</i> argument to display information for a specified tunnel.
Step 3	ping [<i>protocol</i>] <i>destination</i> Example: Router# ping 10.0.0.1	(Optional) Diagnoses basic network connectivity.
Step 4	show ip route [<i>address</i> [<i>mask</i>]] Example: Router# show ip route 10.0.0.2	(Optional) Displays the current state of the routing table. Note Only the syntax relevant for this task is shown.

Examples

This section provides the following output examples:

- [Sample Output from the show interfaces tunnel Command](#)
- [Sample Output from the ping Command](#)
- [Sample Output from the show ip route Command](#)
- [Sample Output from the ping Command](#)

Sample Output from the show interfaces tunnel Command

This example uses a generic example suitable for both IPv6 manually configured tunnels and IPv6 over IPv4 GRE tunnels. In the example, two routers are configured to be endpoints of a tunnel. Router A has Ethernet interface 0/0 configured as tunnel interface 0 with an IPv4 address of 10.0.0.1 and an IPv6 prefix of 2001:0DB8:1111:2222::1/64. Router B has Ethernet interface 0/0 configured as tunnel interface 1 with an IPv4 address of 10.0.0.2 and an IPv6 prefix of 2001:0DB8:1111:2222::2/64. To verify that the tunnel source and destination addresses are configured, use the **show interfaces tunnel** command on Router A.

```
RouterA# show interfaces tunnel 0
```

```
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 10.0.0.1 (Ethernet0/0), destination 10.0.0.2, fastswitch TTL 255
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
```

```

Tunnel TTL 255
Checksumming of packets disabled, fast tunneling enabled
Last input 00:00:14, output 00:00:04, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue :0/0 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
  4 packets input, 352 bytes, 0 no buffer
    Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
  8 packets output, 704 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 output buffer failures, 0 output buffers swapped out

```

Sample Output from the ping Command

To check that the local endpoint is configured and working, use the **ping** command on Router A:

```
RouterA# ping 2001:0DB8:1111:2222::2
```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 2001:0DB8:1111:2222::2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms

```

Sample Output from the show ip route Command

To check that a route exists to the remote endpoint address, use the **show ip route** command:

```
RouterA# show ip route 10.0.0.2
```

```

Routing entry for 10.0.0.0/24
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Routing Descriptor Blocks:
    * directly connected, via Ethernet0/0
      Route metric is 0, traffic share count is 1

```

Sample Output from the ping Command

To check that the remote endpoint address is reachable, use the **ping** command on Router A.



Note

The remote endpoint address may not be reachable using the **ping** command because of filtering, but the tunnel traffic may still reach its destination.

```
RouterA# ping 10.0.0.2
```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/21/28 ms

```

To check that the remote IPv6 tunnel endpoint is reachable, use the **ping** command again on Router A. The same note on filtering also applies to this example.

```
RouterA# ping 1::2
```

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1::2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/20 ms

```


These steps may be repeated at the other endpoint of the tunnel.

Configuration Examples for Implementing Tunneling for IPv6

This section provides the following configuration examples:

- [Configuring Manual IPv6 Tunnels: Example, page 17](#)
- [Configuring GRE Tunnels: Example, page 17](#)
- [Configuring CTunnels in GRE mode to Carry IPv6 Packets in CLNS: Example, page 19](#)
- [Configuring 6to4 Tunnels: Example, page 20](#)
- [Configuring IPv4-Compatible IPv6 Tunnels: Example, page 20](#)
- [Configuring ISATAP Tunnels: Example, page 21](#)

Configuring Manual IPv6 Tunnels: Example

The following example configures a manual IPv6 tunnel between router A and router B. In the example, tunnel interface 0 for both router A and router B is manually configured with a global IPv6 address. The tunnel source and destination addresses are also manually configured.

Router A Configuration

```
interface ethernet 0
 ip address 192.168.99.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::3/127
 tunnel source ethernet 0
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

Router B Configuration

```
interface ethernet 0
 ip address 192.168.30.1 255.255.255.0

interface tunnel 0
 ipv6 address 3ffe:b00:c18:1::2/127
 tunnel source ethernet 0
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

Configuring GRE Tunnels: Example

This section provides the following configuration examples:

- [GRE Tunnel Running IS-IS and IPv6 Traffic: Example, page 18](#)
- [Tunnel Destination Address for IPv6 Tunnel: Example, page 18](#)

GRE Tunnel Running IS-IS and IPv6 Traffic: Example

The following example configures a GRE tunnel running both IS-IS and IPv6 traffic between router A and router B:

Router A Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::1/64
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 10.0.0.2
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.1 255.255.255.0
!
router isis
 net 49.0000.0000.000a.00
```

Router B Configuration

```
ipv6 unicast-routing
clns routing
!
interface tunnel 0
 no ip address
 ipv6 address 2001:0DB8:1111:2222::2/64
 ipv6 router isis
 tunnel source Ethernet 0/0
 tunnel destination 10.0.0.1
 tunnel mode gre ipv6
!
interface Ethernet0/0
 ip address 10.0.0.2 255.255.255.0
!
router isis
 net 49.0000.0000.000b.00
 address-family ipv6
 redistribute static
 exit-address-family
```

Tunnel Destination Address for IPv6 Tunnel: Example

The following example shows how to configure the tunnel destination address for GRE tunneling of IPv6 packets:

```
Router(config)# interface Tunnel0
Router(config-if)# no ip address
Router(config-if)# ipv6 router isis
Router(config-if)# tunnel source Ethernet 0/0
Router(config-if)# tunnel destination 2001:0DB8:1111:2222::1/64
Router(config-if)# tunnel mode gre ipv6
Router(config-if)# exit
!
Router(config)# interface Ethernet0/0
Router(config-if)# ip address 10.0.0.1 255.255.255.0
Router(config-if)# exit
```

```

!
Router(config)# ipv6 unicast-routing

Router(config)# router isis
Router(config)# net 49.0000.0000.000a.00

```

Configuring CTunnels in GRE mode to Carry IPv6 Packets in CLNS: Example

The following example configures a GRE CTunnel running both IS-IS and IPv6 traffic between router A and router B in a CLNS network. The **ctunnel mode gre** command allows tunneling between Cisco and third-party networking devices and carries both IPv4 and IPv6 traffic.

The **ctunnel mode gre** command provides a method of tunneling compliant with RFC 3147 and should allow tunneling between Cisco equipment and third-party networking devices.

Router A

```

ipv6 unicast-routing

clns routing

interface ctunnel 102

  ipv6 address 2001:0DB8:1111:2222::1/64
  ctunnel destination 49.0001.2222.2222.2222.00
  ctunnel mode gre

interface Ethernet0/1
  clns router isis

router isis
  net 49.0001.1111.1111.1111.00

```

Router B

```

ipv6 unicast-routing

clns routing

interface ctunnel 201
  ipv6 address 2001:0DB8:1111:2222::2/64
  ctunnel destination 49.0001.1111.1111.1111.00
  ctunnel mode gre

interface Ethernet0/1
  clns router isis

router isis
  net 49.0001.2222.2222.2222.00

```

To turn off the GRE mode and restore the CTunnel to the default Cisco encapsulation routing only between endpoints on Cisco equipment, use either the **no ctunnel mode** command or the **ctunnel mode cisco** command. The following example shows the same configuration modified to transport only IPv4 traffic.

Configuring 6to4 Tunnels: Example

The following example configures a 6to4 tunnel on a border router in an isolated IPv6 network. The IPv4 address is 192.168.99.1, which translates to the IPv6 prefix of 2002:c0a8:6301::/48. The IPv6 prefix is subnetted into 2002:c0a8:6301::/64 for the tunnel interface: 2002:c0a8:6301:1::/64 for the first IPv6 network, and 2002:c0a8:6301:2::/64 for the second IPv6 network. The static route ensures that any other traffic for the IPv6 prefix 2002::/16 is directed to tunnel interface 0 for automatic tunneling.

```
interface Ethernet0
  description IPv4 uplink
  ip address 192.168.99.1 255.255.255.0
!
interface Ethernet1
  description IPv6 local network 1
  ipv6 address 2002:c0a8:6301:1::1/64
!
interface Ethernet2
  description IPv6 local network 2
  ipv6 address 2002:c0a8:6301:2::1/64
!
interface Tunnel0
  description IPv6 uplink
  no ip address
  ipv6 address 2002:c0a8:6301::1/64
  tunnel source Ethernet 0
  tunnel mode ipv6ip 6to4
!
ipv6 route 2002::/16 tunnel 0
```

Configuring IPv4-Compatible IPv6 Tunnels: Example

The following example configures an IPv4-compatible IPv6 tunnel that allows Border Gateway Protocol (BGP) to run between a number of routers without having to configure a mesh of manual tunnels. Each router has a single IPv4-compatible tunnel, and multiple BGP sessions can run over each tunnel, one to each neighbor. Ethernet interface 0 is used as the tunnel source. The tunnel destination is automatically determined by the IPv4 address in the low-order 32 bits of an IPv4-compatible IPv6 address. Specifically, the IPv6 prefix 0:0:0:0:0 is concatenated to an IPv4 address (in the format 0:0:0:0:0:A.B.C.D or ::A.B.C.D) to create the IPv4-compatible IPv6 address. Ethernet interface 0 is configured with a global IPv6 address and an IPv4 address (the interface supports both the IPv6 and IPv4 protocol stacks).

Multiprotocol BGP is used in the example to exchange IPv6 reachability information with the peer 10.67.0.2. The IPv4 address of Ethernet interface 0 is used in the low-order 32 bits of an IPv4-compatible IPv6 address and is also used as the next-hop attribute. Using an IPv4-compatible IPv6 address for the BGP neighbor allows the IPv6 BGP session to be automatically transported over an IPv4-compatible tunnel.

```
interface tunnel 0
  tunnel source Ethernet 0
  tunnel mode ipv6ip auto-tunnel

interface ethernet 0
  ip address 10.27.0.1 255.255.255.0
  ipv6 address 3000:2222::1/64

router bgp 65000
  no synchronization
  no bgp default ipv4-unicast
```

```
neighbor ::10.67.0.2 remote-as 65002

address-family ipv6
neighbor ::10.67.0.2 activate
neighbor ::10.67.0.2 next-hop-self
network 2001:2222:d00d:b10b::/64
```

Configuring ISATAP Tunnels: Example

The following example shows the tunnel source defined on Ethernet 0 and the **tunnel mode** command used to configure the ISATAP tunnel. Router advertisements are enabled to allow client autoconfiguration.

```
ipv6 unicast-routing
interface tunnel 1
 tunnel source ethernet 0
 tunnel mode ipv6ip isatap
 ipv6 address 2001:0DB8::/64 eui-64
 no ipv6 nd suppress-ra
 exit
```

Where to Go Next

- If you have configured an automatic 6to4 tunnel you can design your IPv6 network around the /48 6to4 prefix you have created from your IPv4 address.
- If you want to implement IPv6 routing protocols, refer to the [Implementing RIP for IPv6](#), [Implementing IS-IS for IPv6](#), [Implementing OSPF for IPv6](#), or [Implementing Multiprotocol BGP for IPv6](#) module.
- If you want to implement security features for your IPv6 network, refer to the [Implementing IPsec in IPv6 Security](#) module.

Additional References

The following sections provide references related to the Implementing Tunneling for IPv6 feature.

Related Documents

Related Topic	Document Title
IPsec VTIs	Implementing IPsec in IPv6 Security
IPv6 supported feature list	Start Here: Cisco IOS Software Release Specifics for IPv6 Features
CLNS tunnels	Cisco IOS ISO CLNS Configuration Guide
IPv6 commands: complete command syntax, command mode, defaults, usage guidelines, and examples	Cisco IOS IPv6 Command Reference

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2473	<i>Generic Packet Tunneling in IPv6 Specification</i>
RFC 2893	<i>Transition Mechanisms for IPv6 Hosts and Routers</i>
RFC 3056	<i>Connection of IPv6 Domains via IPv4 Clouds</i>
RFC 4214	<i>Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)</i>

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS IPv6 Command Reference* at http://www.cisco.com/en/US/docs/ios/ipv6/command/reference/ipv6_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at <http://tools.cisco.com/Support/CLILookup> or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **ctunnel mode**
- **show ipv6 tunnel**
- **tunnel destination**
- **tunnel mode**
- **tunnel mode ipv6ip**
- **tunnel source**

Feature Information for Implementing Tunneling for IPv6

Table 4 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(2)T or a later release appear in the table.

For information on a feature in this technology that is not documented here, see the [Start Here: Cisco IOS Software Release Specifies for IPv6 Features](#) roadmap.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to <http://www.cisco.com/go/cfn>. An account on Cisco.com is not required.



Note

Table 4 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 4 Feature Information for Implementing Tunneling for IPv6

Feature Name	Releases	Feature Information
IPv6 tunneling: manually configured IPv6 over IPv4 tunnels	12.0(23)S ¹ 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T	A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The following sections provide information about this feature: <ul style="list-style-type: none"> Overlay Tunnels for IPv6, page 2 IPv6 Manually Configured Tunnels, page 4 Configuring Manual IPv6 Tunnels, page 7 Configuring Manual IPv6 Tunnels: Example, page 17
CEFv6 Switching for 6to4 Tunnels	12.2(28)SB 12.2(25)SG 12.2(33)SRA 12.2(18)SXE 12.2(12)T 12.4 Cisco IOS XE Release 2.1	Cisco Express Forwarding switching can be used for IPv6 manually configured tunnels. The following sections provide information about this feature: <ul style="list-style-type: none"> IPv6 Manually Configured Tunnels, page 4

Table 4 **Feature Information for Implementing Tunneling for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 tunneling: automatic 6to4 tunnels	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(18)SXE 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	An automatic 6to4 tunnel allows isolated IPv6 domains to be connected over an IPv4 network to remote IPv6 networks. The following sections provide information about this feature: <ul style="list-style-type: none"> • Automatic 6to4 Tunnels, page 5 • Configuring Automatic 6to4 Tunnels, page 10
IPv6 tunneling: automatic IPv4-compatible tunnels	12.0(22)S 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(18)SXE 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	Automatic IPv4-compatible tunnels use IPv4-compatible IPv6 addresses. The following sections provide information about this feature: <ul style="list-style-type: none"> • Overlay Tunnels for IPv6, page 2 • Automatic IPv4-Compatible IPv6 Tunnels, page 5 • Configuring IPv4-Compatible IPv6 Tunnels, page 12 • Configuring IPv4-Compatible IPv6 Tunnels: Example, page 20
IPv6 tunneling: manually configured IPv6 over IPv4 tunnels	12.0(23)S ¹ 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(17a)SX1 12.2(2)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	A manually configured tunnel is equivalent to a permanent link between two IPv6 domains over an IPv4 backbone. The following sections provide information about this feature: <ul style="list-style-type: none"> • Restrictions for Implementing Tunneling for IPv6, page 2 • Overlay Tunnels for IPv6, page 2 • IPv6 Manually Configured Tunnels, page 4 • Configuring Manual IPv6 Tunnels, page 7 • Configuring Manual IPv6 Tunnels: Example, page 17

Table 4 **Feature Information for Implementing Tunneling for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 tunneling: IPv6 over IPv4 GRE tunnels	12.0(22)S ² 12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(17a)SX1 12.2(4)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	GRE tunnels are links between two points, with a separate tunnel for each link. The tunnels are not tied to a specific passenger or transport protocol, but in this case carry IPv6 as the passenger protocol with the GRE as the carrier protocol and IPv4 or IPv6 as the transport protocol. The following sections provide information about this feature: <ul style="list-style-type: none"> • Overlay Tunnels for IPv6, page 2 • GRE/IPv4 Tunnel Support for IPv6 Traffic, page 4 • Configuring GRE IPv6 Tunnels, page 8 • Configuring GRE Tunnels: Example, page 17
IPv6 tunneling: IPv6 over UTI using a tunnel line card ³	12.0(23)S ¹	IPv6 supports this feature.
IPv6 tunneling: ISATAP tunnel support	12.2(14)S 12.2(28)SB 12.2(33)SRA 12.2(17a)SX1 12.2(15)T 12.3 12.3(2)T 12.4 12.4(2)T Cisco IOS XE Release 2.1	ISATAP is an automatic overlay tunneling mechanism that uses the underlying IPv4 network as a NBMA link layer for IPv6. The following sections provide information about this feature: <ul style="list-style-type: none"> • Overlay Tunnels for IPv6, page 2 • ISATAP Tunnels, page 6 • Configuring ISATAP Tunnels, page 13 • Configuring ISATAP Tunnels: Example, page 21
IPv6 tunneling: IPv4 over IPv6 tunnels	12.2(30)S 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T	IPv6 supports this feature The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Manually Configured Tunnels, page 4 • Configuring Manual IPv6 Tunnels, page 7
IPv6 tunneling: IPv6 over IPv6 tunnels	12.2(30)S 12.3(7)T 12.4 12.4(2)T	IPv6 supports this feature The following sections provide information about this feature: <ul style="list-style-type: none"> • IPv6 Manually Configured Tunnels, page 4 • Configuring Manual IPv6 Tunnels, page 7

Table 4 **Feature Information for Implementing Tunneling for IPv6 (continued)**

Feature Name	Releases	Feature Information
IPv6 tunneling: IP over IPv6 GRE tunnels	12.2(30)S 12.3(7)T 12.4 12.4(2)T	GRE tunnels are links between two points, with a separate tunnel for each link. The following sections provide information about this feature: <ul style="list-style-type: none"> • Overlay Tunnels for IPv6, page 2 • GRE/IPv4 Tunnel Support for IPv6 Traffic, page 4 • Configuring GRE IPv6 Tunnels, page 8
IPv6 tunneling: IPv6 GRE tunnels in CLNS networks	12.2(25)S 12.2(28)SB 12.2(33)SRA 12.3(7)T 12.4 12.4(2)T	GRE tunneling of IPv4 and IPv6 packets through CLNS networks enables Cisco CTunnels to interoperate with networking equipment from other vendors. The following sections provide information about this feature: <ul style="list-style-type: none"> • Overlay Tunnels for IPv6, page 2 • GRE/CLNS Tunnel Support for IPv4 and IPv6 Packets, page 5 • Configuring CTunnels in GRE mode to Carry IPv6 Packets in CLNS: Example, page 19

1. In Cisco IOS Release 12.0(23)S, the Cisco 12000 series Internet router provides enhanced performance for IPv6 manually configured tunnels by processing traffic on the line card.
2. IPv6 over IPv4 GRE tunnels are not supported on the Cisco 12000 series Internet router.
3. Feature is supported on the Cisco 12000 series Internet router only.

CCDE, CCENT, Cisco Eos, Cisco HealthPresence, the Cisco logo, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, Cisco TelePresence, Cisco WebEx, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn and Cisco Store are service marks; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MeetingPlace Chime Sound, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0812R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

© 2001–2009 Cisco Systems, Inc. All rights reserved.

