IP6FD

IPv6 Fundamentals, Design, and Deployment

Volume 1

Version 3.0

Student Guide

Text Part Number: 97-2945-01

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Students, this letter describes important course evaluation access information!

Welcome to Cisco Systems Learning. Through the Cisco Learning Partner Program, Cisco Systems is committed to bringing you the highest-quality training in the industry. Cisco learning products are designed to advance your professional goals and give you the expertise you need to build and maintain strategic networks.

Cisco relies on customer feedback to guide business decisions; therefore, your valuable input will help shape future Cisco course curricula, products, and training offerings. We would appreciate a few minutes of your time to complete a brief Cisco online course evaluation of your instructor and the course materials in this student kit. On the final day of class, your instructor will provide you with a URL directing you to a short post-course evaluation. If there is no Internet access in the classroom, please complete the evaluation within the next 48 hours or as soon as you can access the web.

On behalf of Cisco, thank you for choosing Cisco Learning Partners for your Internet technology training.

Sincerely,

Cisco Systems Learning

Table of Contents

Volume 1

Course Introduction

Overview

The *IPv6 Fundamentals, Design, and Deployment* (IP6FD) 3.0 course is an instructor-led course presented by Cisco training partners to their end-user customers. This five-day course aims at enabling learners to study and configure Cisco IOS Software IP version 6 (IPv6) features. The course is a technology course covering IPv6 design and implementation topics. It provides an overview of IPv6 technologies, briefly covers history of IPv6, describes IPv6 operations, addressing, routing, services, transition, and deployment of IPv6 in enterprise networks. The course also includes case studies useful for deployment scenarios.

Learner Skills and Knowledge

This subtopic lists the skills and knowledge that learners must possess to benefit fully from the course. The subtopic also includes recommended Cisco learning offerings that learners should first complete to benefit fully from this course.

Learner Skills and Knowledge

- Cisco CCNA® certification:
  - *Interconnecting Cisco Network Devices 1* (ICND1)
  - *Interconnecting Cisco Network Devices 2* (ICND2)
- Understanding of networking and routing (on Cisco CCNP® level, but no formal certification is required).
- Working knowledge of the Microsoft Windows operating system.

Course Goal and Objectives

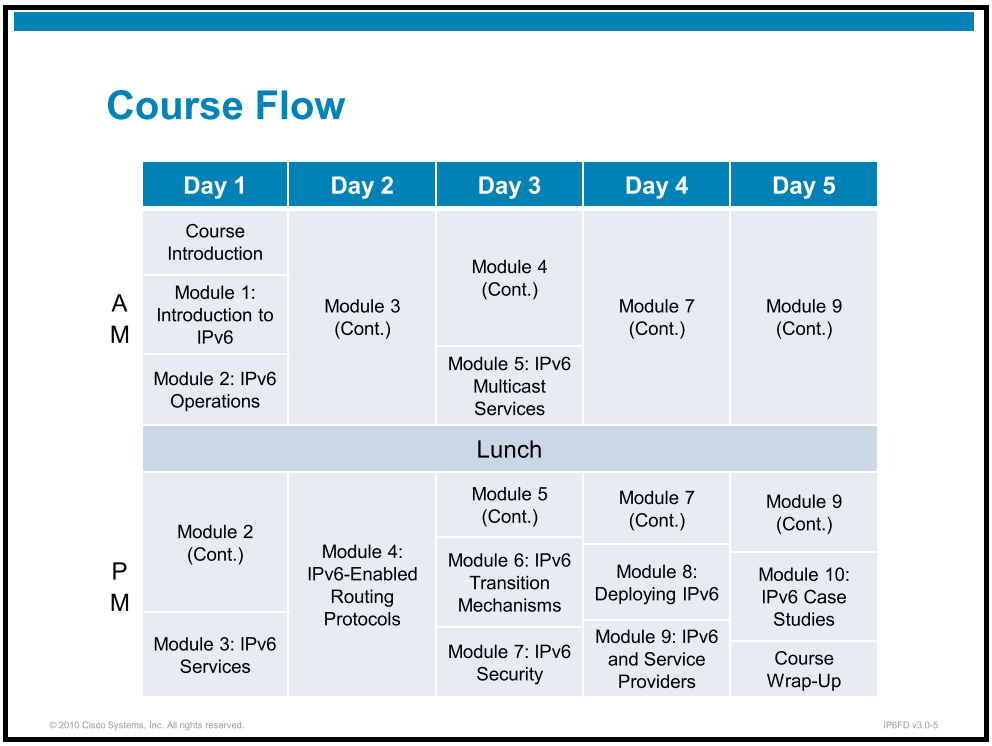
This topic describes the course goal and objectives.

Upon completing this course, you will be able to meet these objectives:

- Describe the factors that led to the development of IPv6, and the possible usages of this new IP structure
- Describe the structure of the IPv6 address format, how IPv6 interacts with link-layer technologies, and how IPv6 is supported in Cisco IOS software
- Describe the nature of changes to DNS and DHCP to support IPv6, and how networks can be renumbered using both services
- Understand the updates to IPv4 routing protocols needed to support IPv6 topologies
- Understand multicast concepts and IPv6 multicast specifics
- Describe IPv6 transition mechanisms and which methods will be most effective in your network
- Describe security issues, how security for IPv6 is different than for IPv4, and emerging practices for IPv6-enabled networks
- Describe the standards bodies that define IPv6 address allocation, as well as one of the leading IPv6 deployment issues, multihoming
- Describe the deployment strategies service providers are facing when deploying IPv6
- Describe case studies for enterprise, service provider, branch, and access networks

Course Flow

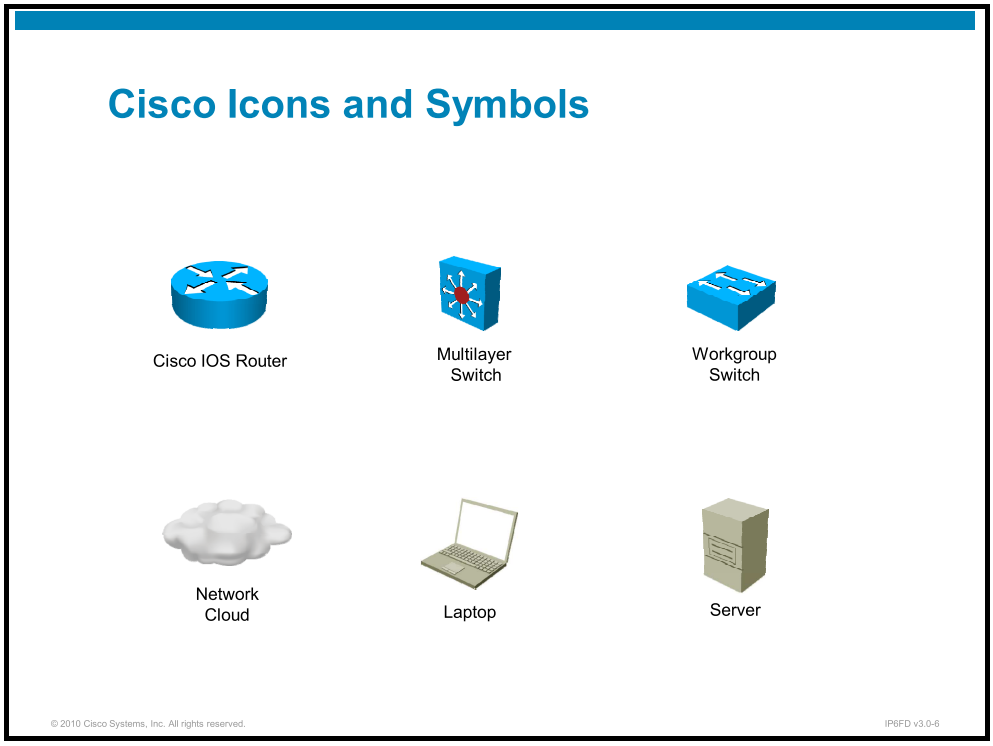
This topic presents the suggested flow of the course materials.



The schedule reflects the recommended structure for this course. This structure allows enough time for the instructor to present the course information and for you to work through the lab activities. The exact timing of the subject materials and labs depends on the pace of your specific class.

Additional References

This topic presents the Cisco icons and symbols that are used in this course, as well as information on where to find additional technical references.



Cisco Glossary of Terms

For additional information on Cisco terminology, refer to the *Cisco Internetworking Terms and Acronyms* glossary of terms at http://docwiki.cisco.com/wiki/Category:Internetworking_Terms_and_Acronyms_%28ITA%29.

Module 1

Introduction to IPv6

Overview

The growth of the Internet and the adoption of networking over the past 20 years are pushing the IP version 4 (IPv4) to the limits of its addressing capacity and its ability for continued growth. To sustain the evolution of the Internet, the Internet Engineering Task Force (IETF) developed a next-generation protocol, IP version 6 (IPv6). This module describes the factors leading toward IPv6 development and compares IPv4 with IPv6.

Module Objectives

Upon completing this module, you will be able to describe the factors that led to the development of IPv6 and possible uses of this new IP structure. This ability includes being able to meet these objectives:

- Describe the history of IPv4 and the rationale for implementing IPv6 to resolve IPv4 addressing and security issues
- Explain the benefits of addressing with IPv6, and describe how larger IPv6 address sizes facilitate autoconfiguration and aggregation
- Describe market drivers and the rationale to move to IPv6; explain the importance of native IPv6 content and how its availability is a motivation to move

Lesson 1

Explaining the Rationale for IPv6

Overview

The development of IP version 6 (IPv6) is a process that has been under way since 1993. Fearing that the world may exhaust the IP version 4 (IPv4) address space, developers of the next-generation protocol, IPv6, not only remedied the address shortage issue but also enhanced the IP. The process of developing a transition plan will also require justification for adopting IPv6. This lesson describes the history of IP networking, defines the IP address shortage issue, and provides foundational materials for advocating a transition to IPv6.

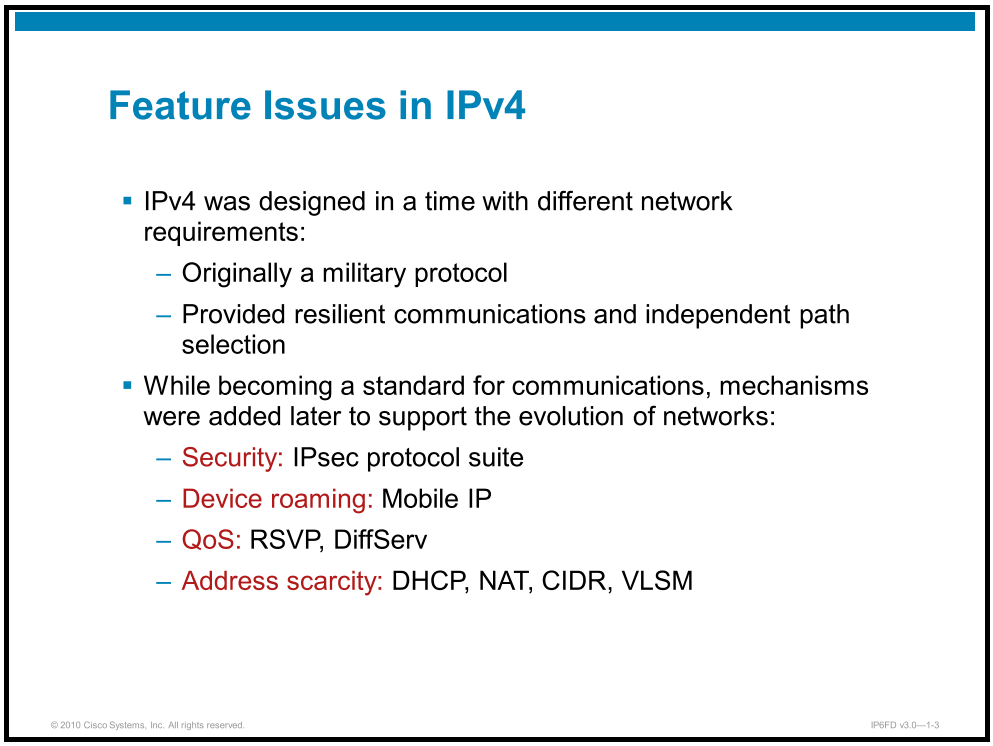
Objectives

Upon completing this lesson, you will be able to describe the history of IPv4 and the rationale for implementing IPv6 to resolve IPv4 addressing and security issues. This ability includes being able to meet these objectives:

- Describe the expansion and adoption of IPv4
- Describe address exhaustion in IPv4
- Describe the rationale for creating a next-generation IP
- Describe why existing solutions, such as NAT, create new issues

IP Address Allocation

This topic describes the expansion and adoption of IPv4.



When the concept of TCP/IP was originally developed by Vint Cerf and Robert Khan under the auspices of the Defense Advanced Research Projects Agency (DARPA), a U.S. Department of Defense (DoD) agency, modern-day network requirements—such as security, quality of service, device autoconfiguration, privacy, and a high level of network device diversity—did not necessarily apply. The DARPA (then known only as ARPA) network was expected to be a closed network that is limited to a few thousand users, mostly scientists from government and academia.

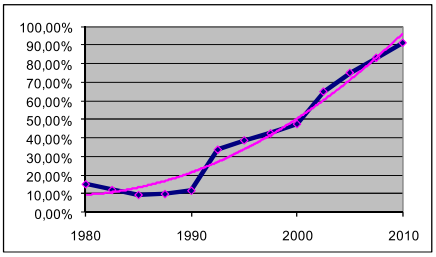
As the use of networking exploded and the Internet emerged, many modern-day network requirements became necessary for the continued growth of the Internet and the large-scale adoption of networking and computers in an enterprise environment.

Nowadays, the IP supports networks all around the world, including businesses, social networks, and governments. To address usage growth, many mechanisms had to be developed to provide security, quality of service (QoS), mobility, and so on.

IP Address Allocation and History

- Address pool usage introduced conservation and optimization efforts, including:
  - PPP and DHCP address sharing, and endpoints behind NAT
  - Subscriber-Grade NAT and Carrier-Grade NAT
  - CIDR and VLSM for address class segmentation and improved routing
- Theoretical limit of 32-bit space is approximately 4.3 billion devices.
- Practical limit of 32-bit space is approximately 250 million devices.
- Neither of these can accommodate one IP address per person.

1981: IPv4 published
1985: ~1/16 of total space used
1990: ~ 1/8 of total space used
1995: ~ 1/3 of total space used
2000: ~ 1/2 of total space used
2003: ~ 2/3 of total space used
2005: ~ 3/4 of total space used
2010: ~ 9/10 of total space used



The American Registry for Internet Numbers (ARIN) has 15 /8s allocated directly from the Internet Address and Numbering Authority (IANA). This does not include the 20 /8s for U.S. corporations (nondefense) or the 64 /8s of the legacy /16s. (These are the address blocks allocated directly from IANA before the formation of ARIN, more commonly referred to as “swamp” space.)

China has approximately one /8, with a larger student population than North America. India has a smaller allocation, but its population is expected to surpass the population of China.

Many allocated Class A addresses are not used and can probably be reclaimed. However, it is much more difficult to obtain new addresses, because they are an increasingly scarce resource. Many conservation efforts, including classless interdomain routing (CIDR), Network Address Translation (NAT), address reclamation, and DHCP, have significantly extended the lifetime of IPv4 and allowed deployment of IP-based networks that would have otherwise been forced to use alternative methods.

Note A /8 is equivalent to a Class A block of IPv4 addresses, or 16 million IP addresses. A /16 is equal to a Class B address block or 65,536 IPv4 addresses.

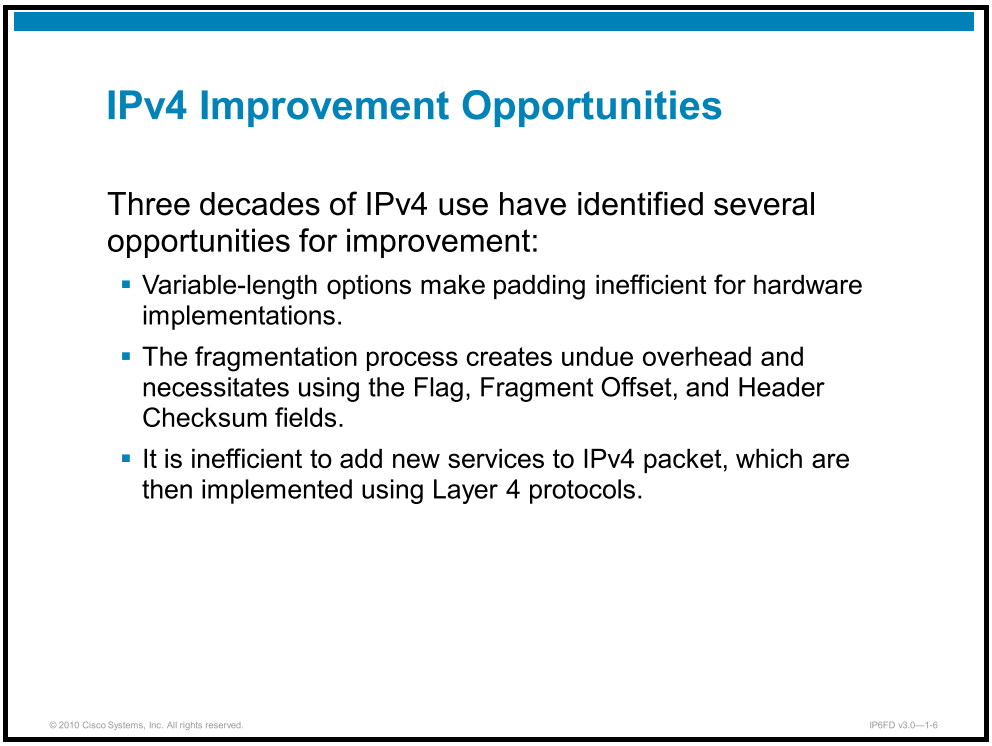
Temporary or semipermanent connections, such as dialup or cable-modem xDSL, which are given either temporary or private IPv4 addresses, prohibit the end user from deploying certain IP-based services. New networks find it increasingly difficult to obtain IPv4 addresses from the registries. With IPv4, technologies such as wireless telephones, personal digital assistants, in-car computers, and home appliances will not be allocated global addresses, because there are too many such devices. In some countries, end users are at a third, fourth, or even fifth layer of NAT networks, limiting them to the most basic of services.

The 32-bit address space of IPv4 accommodates more than 4 billion devices. No addressing scheme is optimal, but to compare different addressing schemes a logarithmic ratio, known as the HD-ratio, is applied to other address spaces—such as the address spaces used in telephone numbers—to compare efficiency of use. IPv4 was not better or worse than others, but the class hierarchy (A, B, C) makes the IPv4 address space less efficient.

Tip The phone system illustrates the inefficiencies in number allocation when the number space is hierarchically arranged. Suppose that two cities need new area codes, from which new local exchanges can be drawn. Each city is assigned one new area code. Soon thereafter, one city grows more quickly and needs more numbers, while the other city experiences no additional growth. Because numbers are grouped into area codes, the responsible organization cannot “borrow” exchange numbers from the “no-growth” city to give to the “growth” city. Exchanges are only valid within a given area code. The inability to easily reallocate phone number assignments between area codes adds inefficiency to the use of the overall number space. Similarly, with IP addresses, some allocations are densely used (most IP addresses assigned) while others are sparsely assigned. The result is that the total “practical” limit of available phone numbers (or IP addresses) is much smaller than the theoretical maximum.

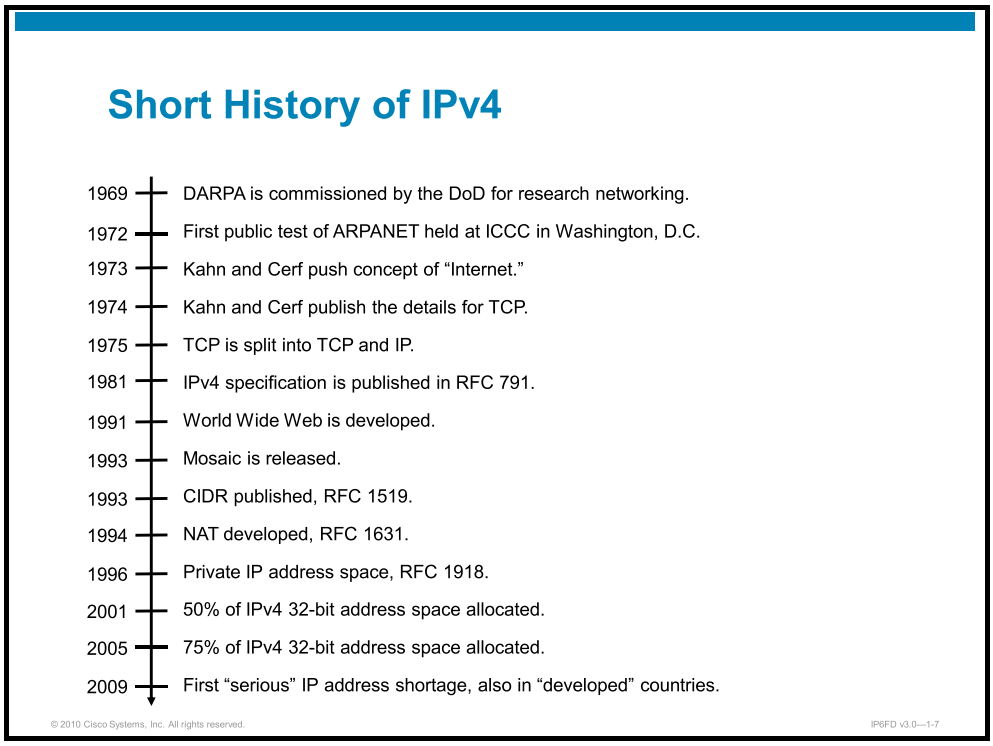
History of IPv4

This topic describes address exhaustion in IPv4.



During the past 30 years, a remarkable number of services have been developed for networks and the Internet. Experience with IP has helped to identify several areas for improvement. During development of the next-generation IP, attempts were made to improve several inefficiencies, such as these:

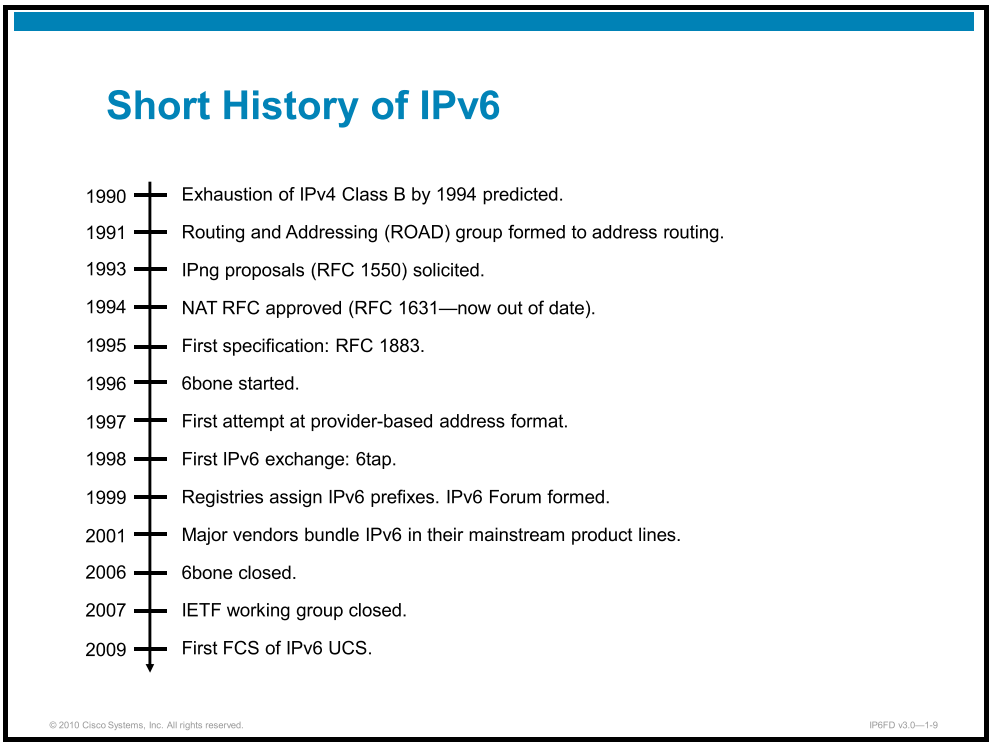
- The IPv4 header may be of variable length. The Options field provides the ability to do source routing, record the packet route, and provide address extension information. (See http://www.iana.org/assignments/ip-parameters for a more detailed list of options.) Since the option data is not a consistent length, padding is often applied to the field. Padding the Options field ensures that the data will start on a 32-bit boundary. Consequently, the field has variable-length headers, creating more overhead in the forwarding process.
- Heterogeneous networks may have many different path maximum transmission units (MTUs). For example, link A on a path may be 1500 bytes, while link B on the same path may be 1300 bytes, necessitating packet fragmentation. This creates overhead for the routing device forwarding the packets, both in fragmentation and in recalculating the checksum.



The concept of networking was introduced by J.C.R. Licklider of the Massachusetts Institute of Technology in August 1962, when he discussed his "Galactic Network" concept. During the 1960s, the idea took form in numerous papers, culminating with the U.S. DoD commissioning DARPA to develop the network concept. Through a series of development iterations, TCP and, eventually, IP were created. IPv4 was standardized in 1981.

Next Generation of IP

This topic describes the rationale for creating a next-generation IP.



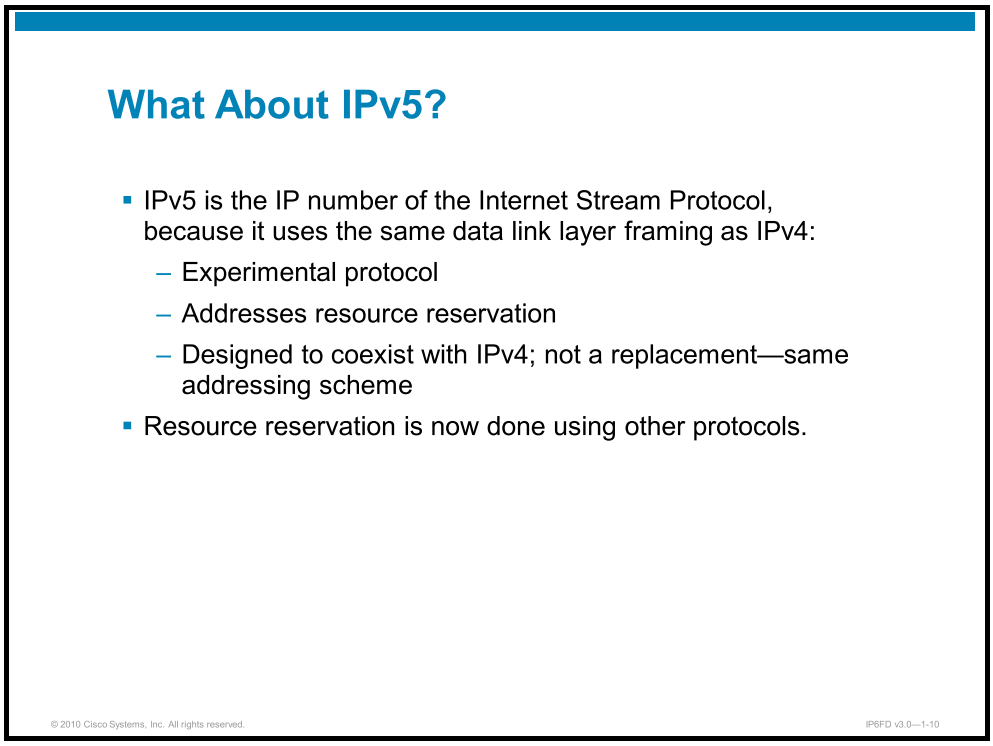
Following the new address space study, the consensus was that there was enough time to develop a new protocol with added functionalities instead of deploying a new protocol that just added larger new addresses. This meant that there was an opportunity to fix problems associated with the IPv4 addressing scheme and create a new protocol for future needs.

The process included collecting requirements from various industries: cable, wireless, electric power utilities, military, corporate networks, service providers, and others. A call for proposals (RFC 1550) was issued, and several proposals were studied.

Of all the proposals, three warranted more extensive analysis and attention: Common Architecture for Next Generation IP (CATNIP), Simple Internet Protocol Plus (SIPP), and TCP and User Datagram Protocol (UDP) with Bigger Addresses (TUBA). The recommended proposal, submitted by the Internet Engineering Task Force (IETF) (RFC 1752), was the SIPP, with a larger address space. The main author of SIPP was Steve Deering. A working group was established, and the first specification came in late 1995 (RFC 1883).

The major milestones of IPv6 are as follows:

- **1996:** A testbed of IPv6, called the 6bone, was started over the Internet. At this time, Cisco already had support for IPv6 on a limited number of Cisco hardware platforms.
- **1997:** A first attempt was made for a provider-based address format.
- **1999:** Registries started to assign IPv6 prefixes to ISPs.
- **1999:** The IPv6 Forum was founded.
- **2000:** Many vendors began to bundle IPv6 into their mainstream product lines.
- **2002:** Cisco introduced IPv6 generic availability in Cisco IOS Software Release 12.2(1)T.



IPv5 was an experimental Resource Reservation Protocol that is intended to provide quality of service (QoS) and defined as the Internet Stream Protocol version 2 (ST2). It is based on the work of Jim Forgie in 1979, as documented in *Internet Experiment Notes* (IEN) 119, and consists of two protocols:

- ST2, for the data transport
- Stream Control Message Protocol (SCMP)

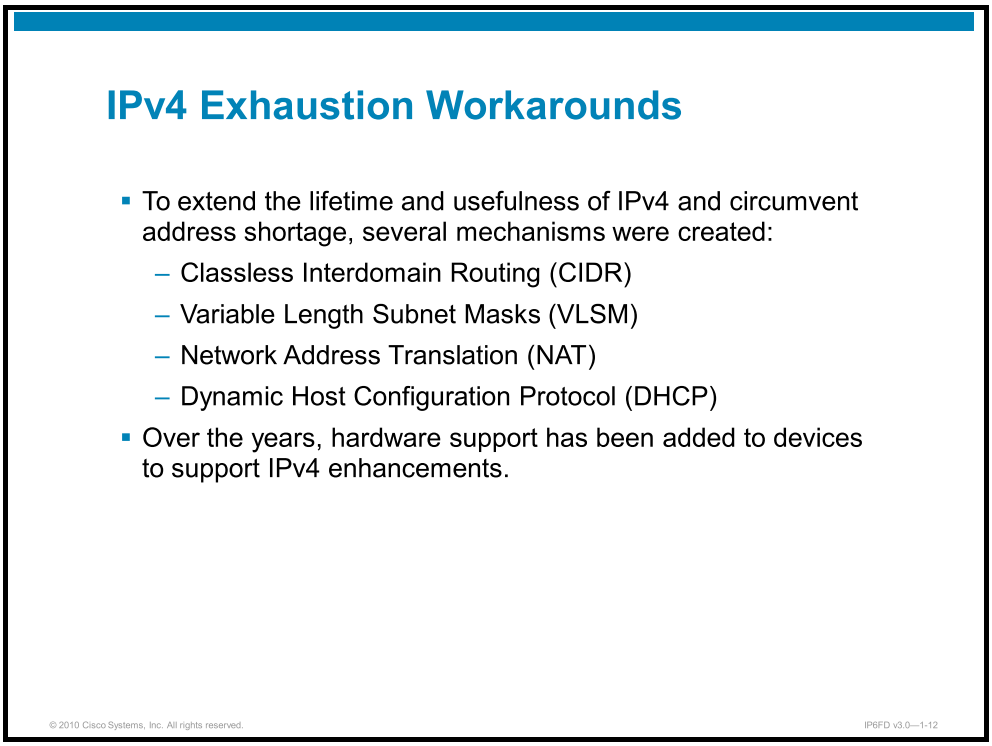
ST2 is designed to coexist with IP on each node. A typical distributed multimedia application would use both protocols: IP for the transfer of traditional data and control information and ST2 for the transfer of real-time data. It uses the same addressing schemes to identify hosts.

ST2 does not replace IP, but it has an IP version number (5) because it uses the same data link layer framing as IPv4. Resource reservation now uses other protocols, for example, Resource Reservation Protocol (RSVP).

ST2 protocol is documented in RFC 1819.

IPv4 Workarounds

This topic describes why existing solutions, such as NAT, create new issues.



IPv4 addresses were originally assigned in several fixed-length blocks, commonly referred to as "classes." Address recipients were allocated Class A, B, or C address blocks, which ranged from approximately 16 million addresses for Class A to 255 addresses for Class C. This allocation policy was not very efficient and resulted in entities receiving many more addresses than they required. As early as 1993, it was stated in RFC 1519 that address depletion was an eventuality.

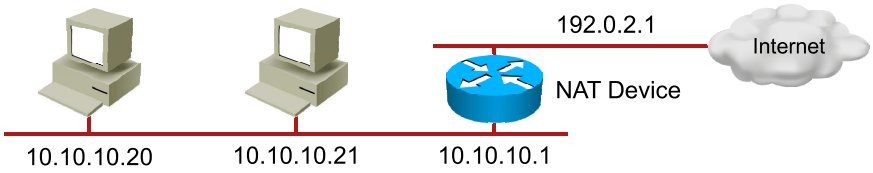
In an effort to allocate addresses more efficiently, CIDR was developed, which allowed the address space to be divided into smaller blocks. In 1994 it was clear that CIDR alone would not provide enough room in the IPv4 address space to allow time for its successor to be developed and deployed. NAT was the proposed temporary workaround. NAT introduced a model in which a device facing outward to the Internet would have a globally routable IPv4 address, while the internal network would be configured with private addresses. These private addresses could never leave the site, so they could be identical in many different enterprise networks. In this way, even large enterprises with thousands of systems could hide behind few routable public networks. As long as the number of client systems behind the NAT (that is, those with hidden private addresses) is large, and those clients are using "well-behaved" applications and are truly clients, not servers, NAT does indeed conserve address space.

Variable-length subnet masks (VLSMs) allow more efficient use of IP addresses, specifically on small segments such as point-to-point serial links. VLSM usage was recommended in 1995 in RFC 1817, and in fact, CIDR and VLSM support is a prerequisite for ISPs to receive additional allocations.

Tip VLSM improves address allocation efficiency at the expense of simplicity. It is simpler to have a standard subnet size—perhaps a /24 or “Class C” subnet. That is inefficient, however, if most subnets in an organization only have 30 or 40 (or even 5) nodes on them. Using VLSM, a 30-node subnet can be allocated a /27, and eight /27 networks can be allocated from a single /24. This complexity must be managed, however, and has other impacts. For example, routing protocols used before VLSM was introduced did not have to carry subnet masks; they assumed that networks had a fixed, consistent netmask. For IPv4, with its limited address space, VLSM has greatly extended the lifetime of the protocol.

Network Address Translation

- NAT provides a way to use a public external address to connect several endpoints with internal addresses behind one or a small number of routable addresses.
- NAT has many implications:
  - Breaks the end-to-end model of IP, especially for inbound connections
  - Mandates that the network keeps the state of connections; however, stateful devices introduce greater security



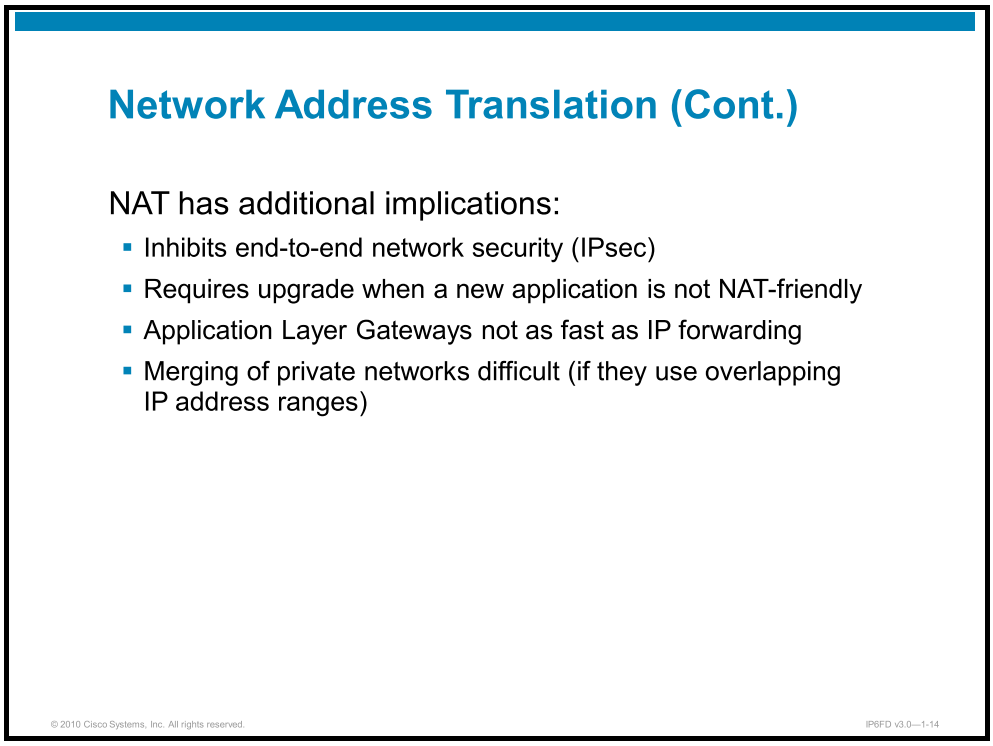
One of the arguments against deploying IPv6 is that NAT will solve the problems of limited address space in IPv4. NAT typically translates packets from a network that uses the private address space (RFC 1918) to the Internet. The use of NAT merely delays the exhaustion of the IPv4 address space by using global addresses for large internal networks.

These address spaces are used within private networks:

- 10.0.0.0/8
- 172.16/12
- 192.168/16

There are several negative implications of using NAT, some of which are identified in RFC 2775 and RFC 2993, as follows:

- NAT breaks the end-to-end model of IP. IP was defined so that underlying layers do not process the connection; only the endpoints process the connection.
- NAT implies that the network needs to keep the state of the connections because it has to remember the translation of addresses and ports. The need to keep the state of the connections in NAT makes fast rerouting difficult in case the NAT device fails or the links near the NAT device fail.

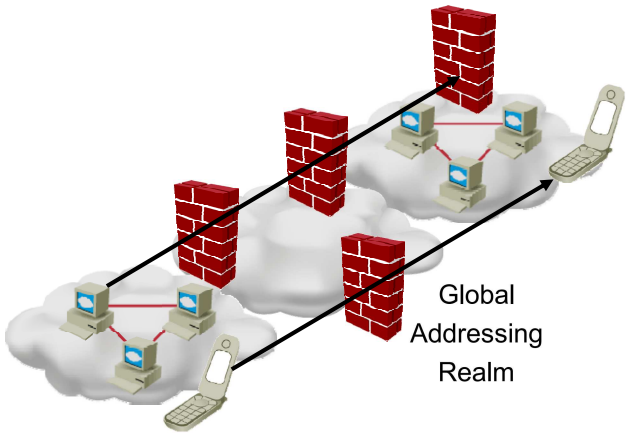


These are other negative implications of NAT:

- NAT inhibits end-to-end network security. To protect the integrity of the IP header by some cryptographic functions, the IP header cannot be changed between the origin of the packet (to protect the integrity of the header) and the final destination (to check the integrity of the received packet). Any translation of parts of a header on the path will break the integrity check. In some cases, you can use adaptations to partially solve the security issue. For example, user hacking.
- When applications are not "NAT-friendly"—which means that, for a specific application, more than just port and address mapping are necessary to forward the packet through the NAT device—NAT has to embed complete knowledge of all the applications to perform correctly. This is especially the case for dynamically allocated ports with rendezvous ports, embedded IP addresses in application protocols, security associations, and so on. Therefore, the NAT device needs to be upgraded each time a new non-NAT-friendly application is deployed. For example, peer-to-peer.
- When different networks use the same private address space, such as 10.0.0.0/8, and they have to merge or connect, there is an address-space collision. Different hosts have the same address, and routing disables reaching the other network. This can be resolved by techniques such as renumbering or Twice NAT. (Twice NAT is the practice of changing both the source and destination address of a packet.) However, these techniques are costly and, later on, increase NAT complications.

NAT Inhibits Access to Internal Servers

- When many internal servers need to be reachable from outside, NAT becomes an important issue.

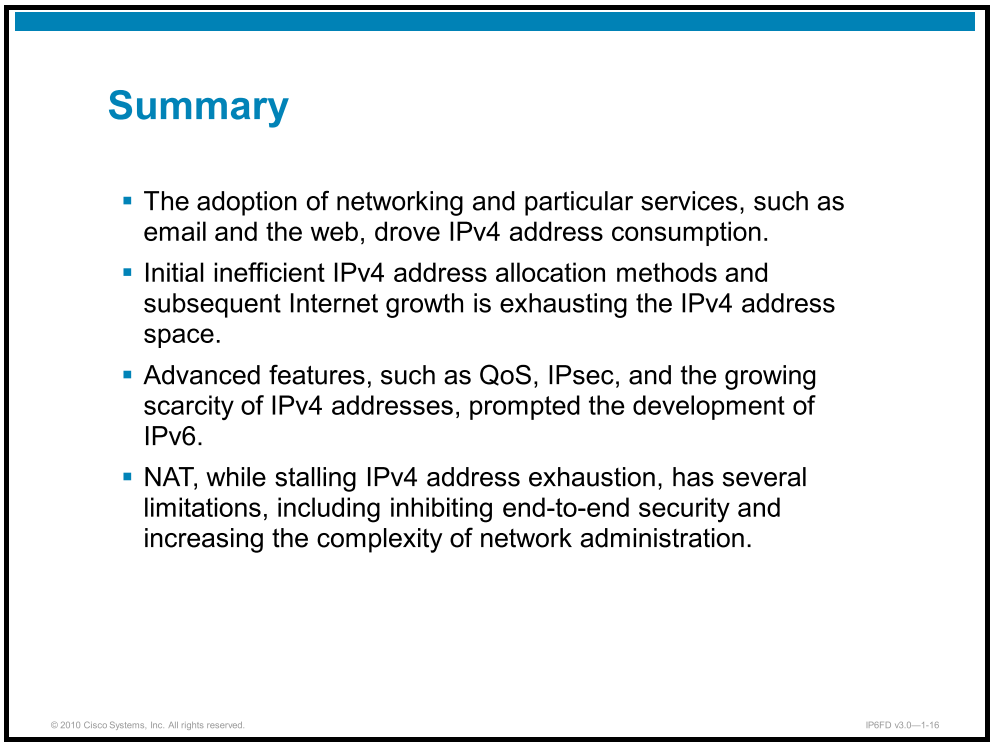


NAT can be useful when there are many devices inside and very few reachable addresses outside. The ratio of internal to reachable must be large to make NAT effective. However, when there are many servers inside, NAT becomes a problem because the same protocol cannot be multiplexed on the same port using the NAT external address. For example, two internal servers using the same port cannot use the same external address without changing the port number.

Each inside server that has to be reachable from the outside will start using one external address. As the number of nodes acting like servers increases (applications running on a node that would make it act like a server, even temporarily), so can the consumption of external IPv4 addresses. With a limited pool of external addresses, NAT effectiveness decreases.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 2

Evaluating IPv6 Features and Benefits

Overview

Similar to any upgrade in technology, there should be enough enhancements and benefits to warrant the time, effort, and cost associated with the upgrade process. IP version 6 (IPv6) provides several improvements over its predecessor and has new features that will help support the continued growth of the Internet. This lesson describes IPv6 features and benefits.

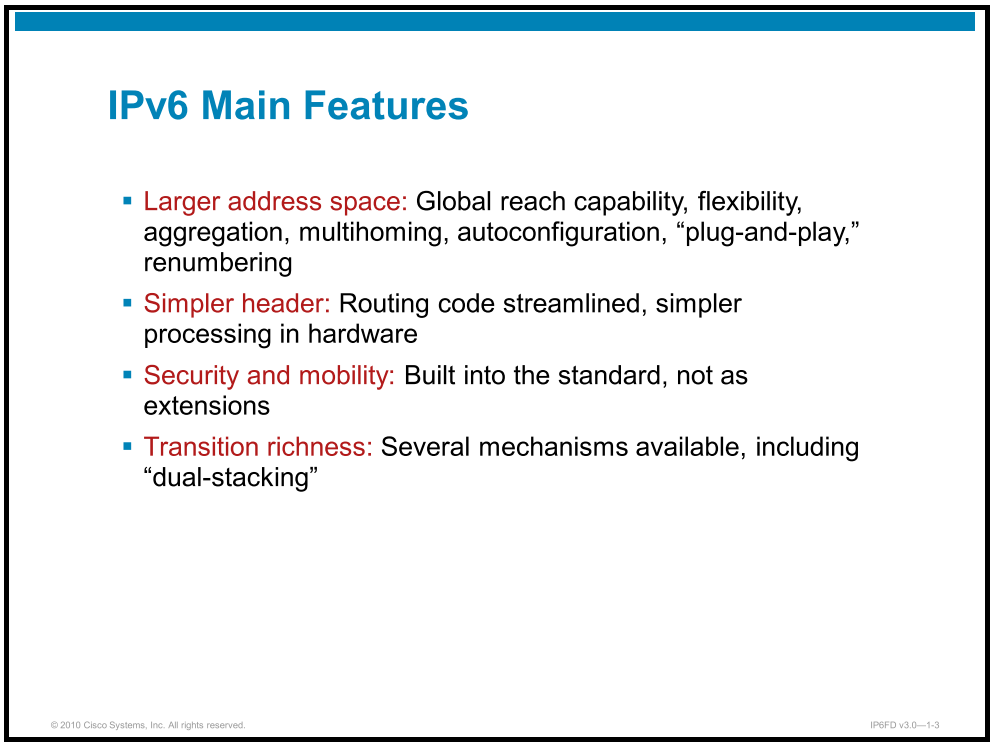
Objectives

Upon completing this lesson, you will be able to explain the benefits of addressing with IPv6 and describe how larger IPv6 address sizes facilitate autoconfiguration and aggregation. This ability includes being able to meet these objectives:

- Describe the features and benefits of IPv6
- Explain the size of an IPv6 address
- Describe how a larger IPv6 address space enables autoconfiguration and aggregation
- Discuss advanced IPv6 features
- Discuss the transition strategies to IPv6

Features and Benefits of IPv6

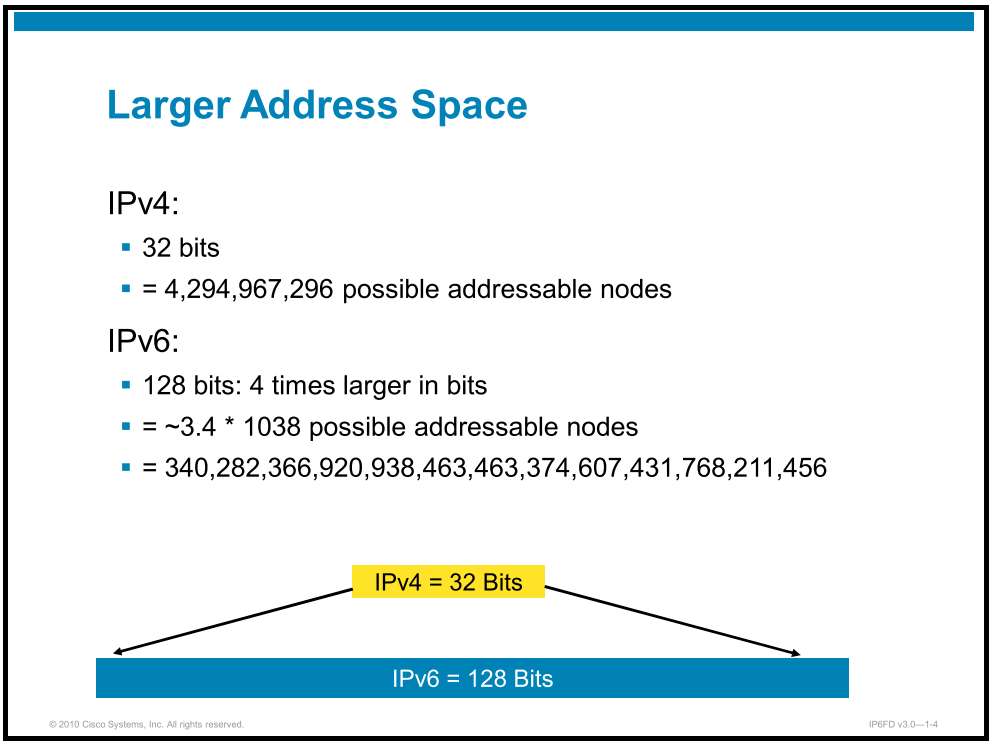
This topic describes the features and benefits of IPv6.



IPv6 includes a number of features that make it attractive for building global-scale, highly effective networks. The larger address space, strict aggregation, and autoconfiguration provide important capabilities.

Streamlined header structures make processing IPv6 packets faster and more efficient for intermediate routers within the network. This is especially true when large numbers of packets are routed in the core of the IPv6 Internet. Features that were not part of the original IPv4 specification, such as security and mobility, are now built into IPv6.

IPv6 also includes a rich set of transition tools to allow an easy, nondisruptive transition over time to IPv6-dominant networks.



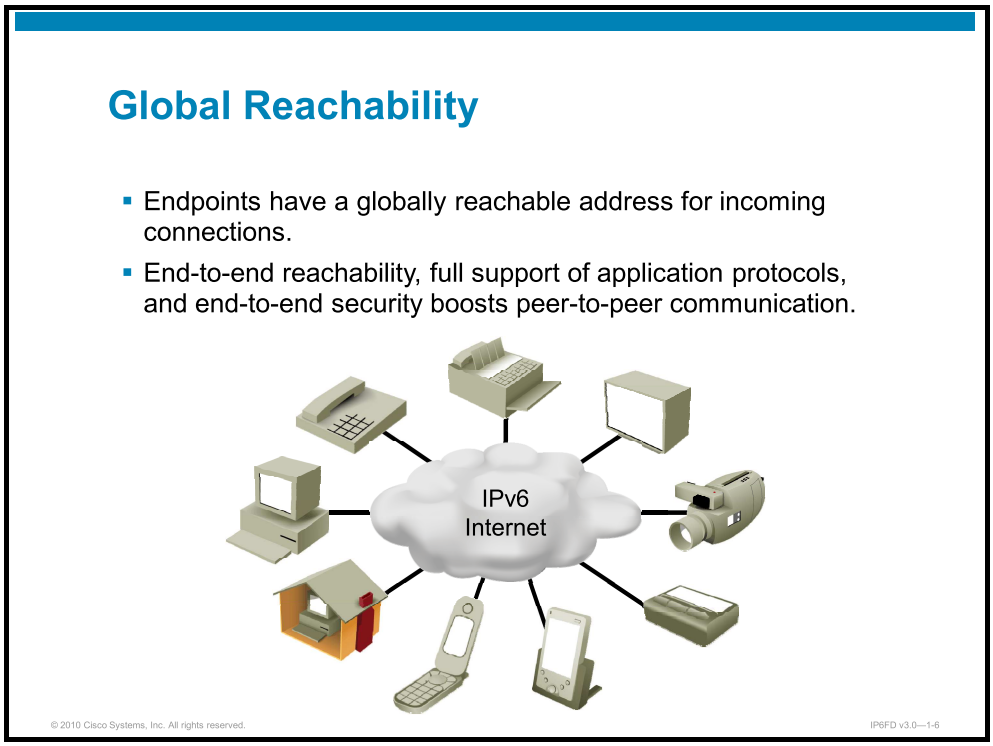
IPv6 increases the number of address bits by a factor of four, from 32 to 128 bits. During the IPv6 design specification, there was a debate about whether to use 64, 128, or 160 bits. The choice of 128 bits was found to be the most appropriate, because it enables a very large number of possible addressable nodes. However, as in any addressing scheme, not all the addresses can be used.

Increasing the number of bits for the address also means an increase in the header size. Since each IP header contains both a source address and a destination address, the size of the header fields containing the addresses is 64 bits for IPv4 and 256 bits for IPv6.

With the upcoming phenomenon of social networks and persons owning multiple devices connected at the same time (mobile phones, home-theater PCs, content servers, and so on), IPv6 is suitable to enable end-to-end communication.

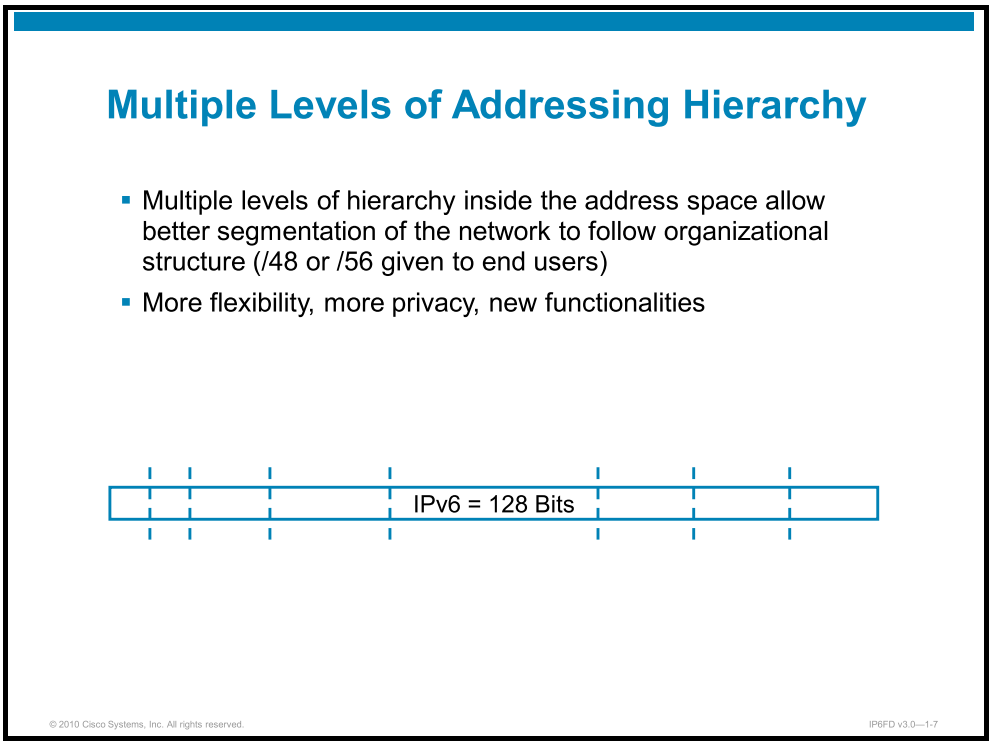
IPv6 Addresses

This topic describes the size of an IPv6 address.



The important issue that motivated the IPv6 effort was the study predicting that address space exhaustion would result from the attempt to give an address to each device on the Internet. By using a much larger address space than IPv4, IPv6 enables the use of a global and reachable address for almost every kind of device: computers, IP phones, IP faxes, TV setup boxes, cameras, pagers, wireless personal digital assistants (PDAs), cell phones, home networking systems, and vehicles. Trying to fit all those devices into the current IPv4 address space is nearly impossible.

A unique address for each device enables the end-to-end reachability that is especially important for telephone calls. Unlike Network Address Translation (NAT) devices, IPv6 enables complete support of application protocols without needing special processing at the edges of the networks. It also enables end-to-end security.



A much larger address space allows the use of multiple levels of hierarchy inside the address space. Each level can help aggregate the traffic and enhance the allocation function. By using multiple levels in the hierarchy, the larger address space permits flexibility and new functionalities, such as the scoping of addresses (site-local, link-local) in the protocol. A flexible addressing architecture is often crucial to a network protocol.

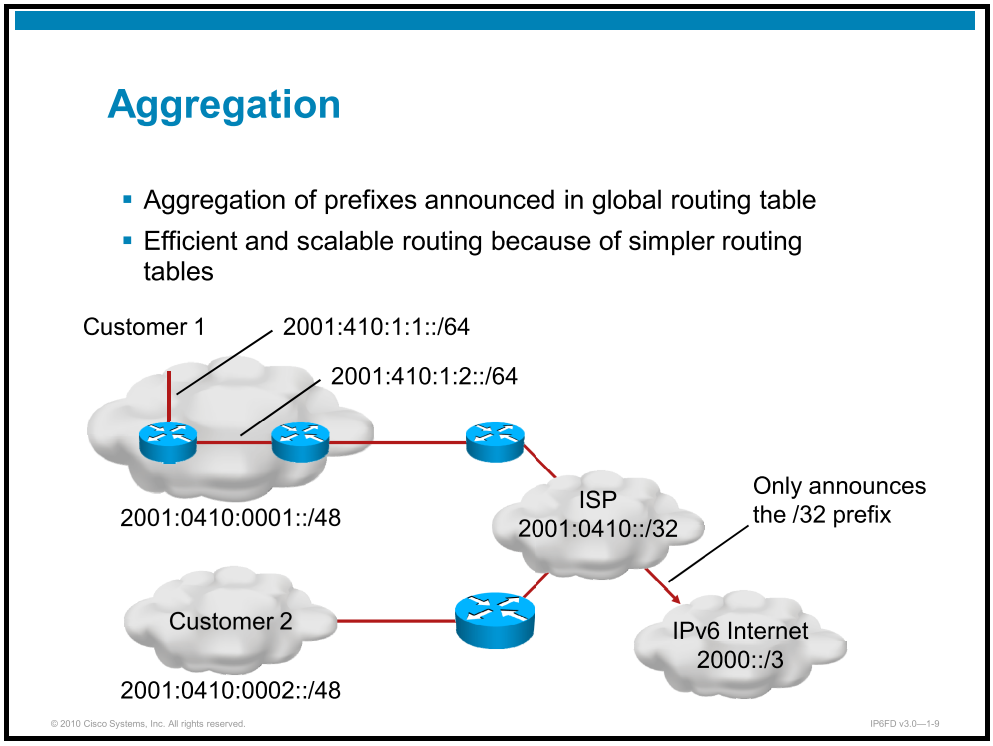
Typical allocations of address space are as follows:

- /32 for a large ISP
- /48 or /56 for a typical enterprise, home, or other place needing more than one network
- /64 for environments where only one network is needed
- /128 for environments where one, and only one, host will need an address

Within the enterprise scenario, it is expected that the first 48 bits are the overall enterprise allocation, which will be split into 16 bits for networks (65,535 networks). Each network will have 64 host bits (primarily for autoconfiguration, not because there would be that many hosts on a single network).

IPv6 Autoconfiguration and Aggregation

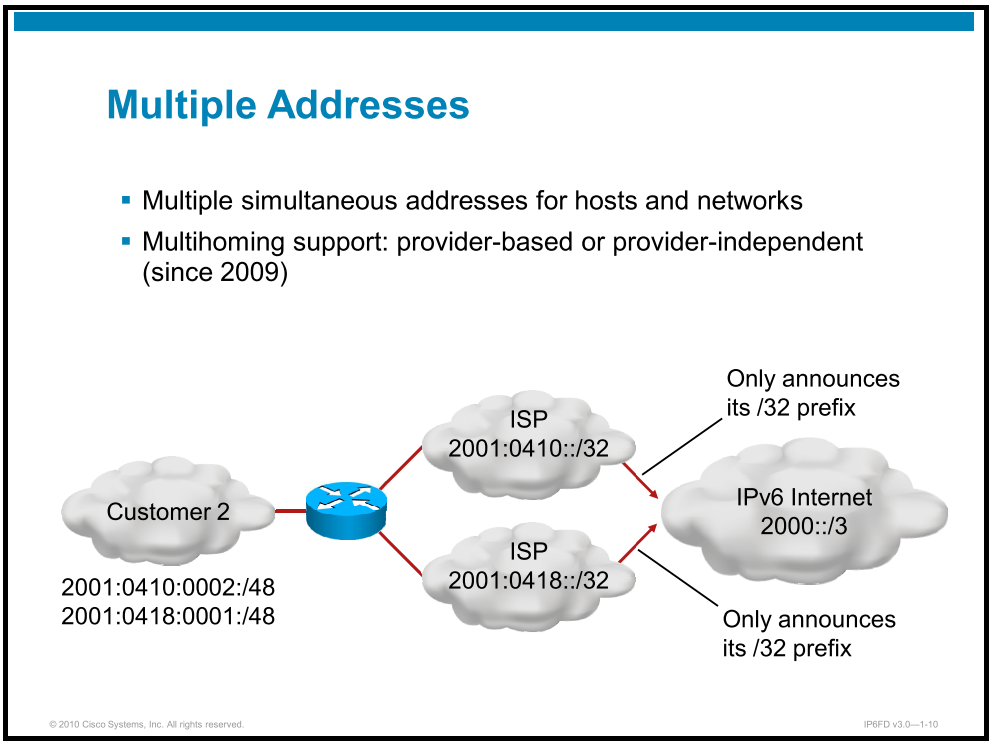
This topic describes how a larger IPv6 address space enables autoconfiguration and aggregation.



A larger address space makes room for large address allocations to service providers and organizations. Having a large enough prefix for the whole network of an organization enables that organization to use only one prefix. On another level, the ISP is able to aggregate all of its customer prefixes into a single prefix and announce this one prefix to the IPv6 Internet.

This aggregation promotes efficient and scalable routing. To connect all kinds of devices and networks on the Internet in the future, scalable routing will be required.

However, this is an oversimplified view. In fact, private peering agreements among all Tier-2 ISPs can undermine complete aggregation. For the core backbone Internet routers, however, a maximum prefix length will be carried (perhaps /32, perhaps something a little longer, but almost certainly not /48). Therefore, route aggregation will be better than it is on the IPv4 Internet.



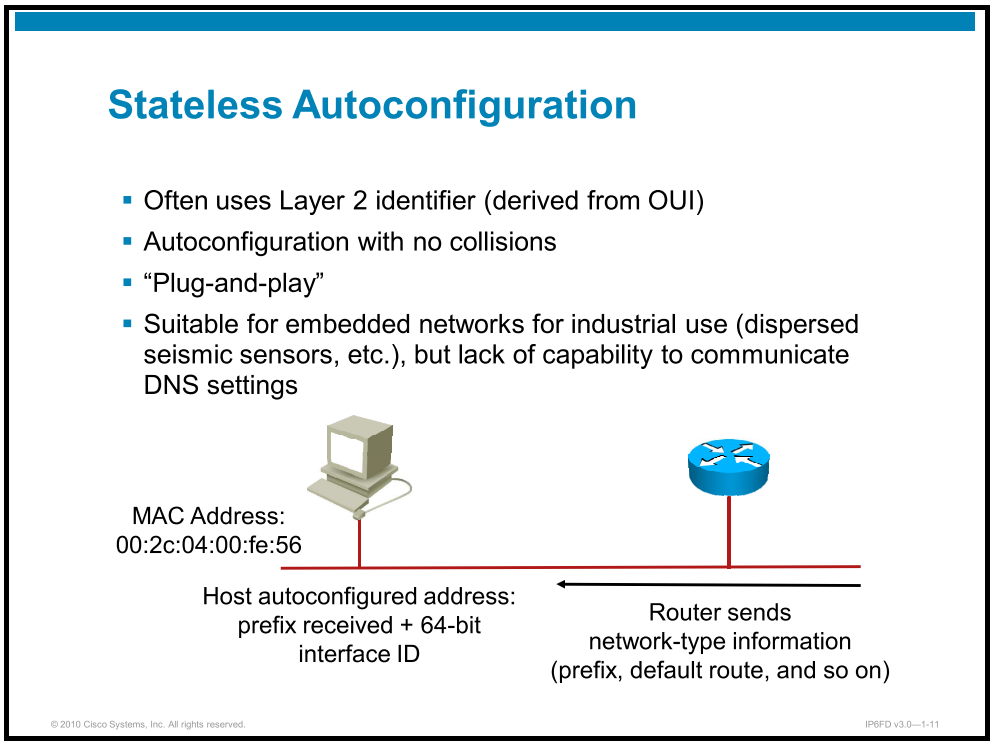
It is not simple to connect a network to multiple providers with IPv4. At the very least, multihoming breaks any kind of aggregation in the global routing table. However, multihoming is desirable for networks that require high availability.

Having a much larger address space, IPv6 enables the use of multiple simultaneous prefixes for a network, allowing multihoming without stressing the global routing table. This situation is not currently possible in IPv4.

The IPv6 model, however, is not without flaws. Having multiple prefixes available to enterprises introduces new challenges. These challenges include overcoming the restrictive nature of aggregation for global networks, detecting when a prefix is no longer being routed (due to an ISP outage), added complexity to the Domain Name Service (DNS) infrastructure, and choosing which prefix to source packets from when multiple, globally routable addresses are available.

Note	There are many ways to make the source address selection, which is covered more fully in *Default Address Selection for IP version 6 (IPv6)* (RFC 3484).

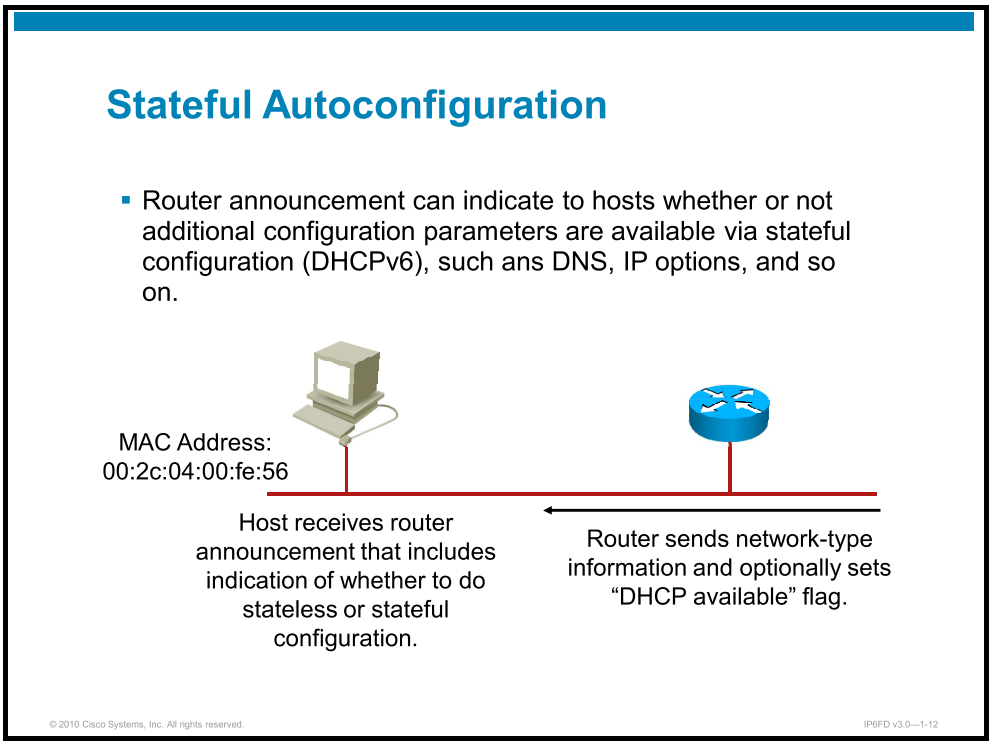
To support “traditional” multihoming using BGP for path selection in the same way as IPv4, Réseaux IP Européens (RIPE) opened the applications (in 2009) for the enterprises to request IPv6 provider-independent address space. This way, the customer advertises the same IPv6 prefix to both service providers, and BGP takes care of customer reachability in the service provider backbones.



Having a much larger address space available, IPv6 engineers designed a way to enable autoconfiguration of the addresses while still keeping the global uniqueness. A router on the local link will send network-type information, such as the prefix of the local link and the default route, to all the nodes on the local link. A host can autoconfigure itself by appending its data link layer address (in a special 64-bit extended unique identifier [EUI]-64 format) to the local link prefix (64 bits). This autoconfiguration results in a complete 128-bit IPv6 address that is usable on the local link and is, most likely, globally unique. To avoid the rare event of address collision, a process is enabled to detect duplicate addresses.

Autoconfiguration enables "plug and play," which connects devices to the network without any configuration and without any stateful servers (such as DHCP servers). Autoconfiguration is an important feature for enabling deployment of new devices on the Internet, such as cell phones, wireless devices, home appliances and networks, and so on.

Autoconfiguration can be accomplished in two ways: stateless—via neighbor discovery and router advertisements—as described above, and stateful, using a DHCPv6 server. The difference between the two is that with the stateful method, a record is kept of which hosts are assigned which addresses. The stateless method maintains no such records.

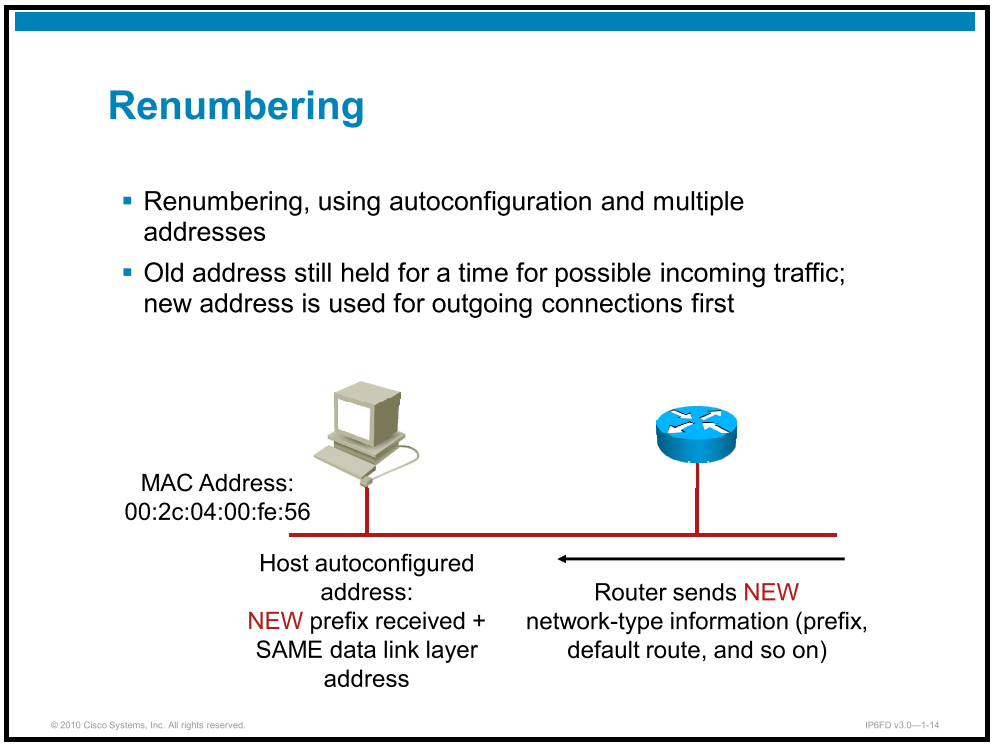


Stateless autoconfiguration is a powerful mechanism for zero-administration configuration. A typical full-featured client, such as an enterprise desktop computer, however, can require a much larger set of initial configuration settings than stateless autoconfiguration can provide. For these applications, there are additional settings, such as DNS servers, Network Time Protocol (NTP) servers, Session Initiation Protocol (SIP) servers, Novell Directory Services, and others.

Hosts use stateless autoconfiguration to receive base information (in some cases) along with an indication of whether or not additional configuration settings are available via a DHCP server. This can include the IP options, in addition to autoconfigure hardware such as IP phones, wireless access points, video endpoints, and so on.

Advanced IPv6 Features

This topic describes advanced IPv6 features.

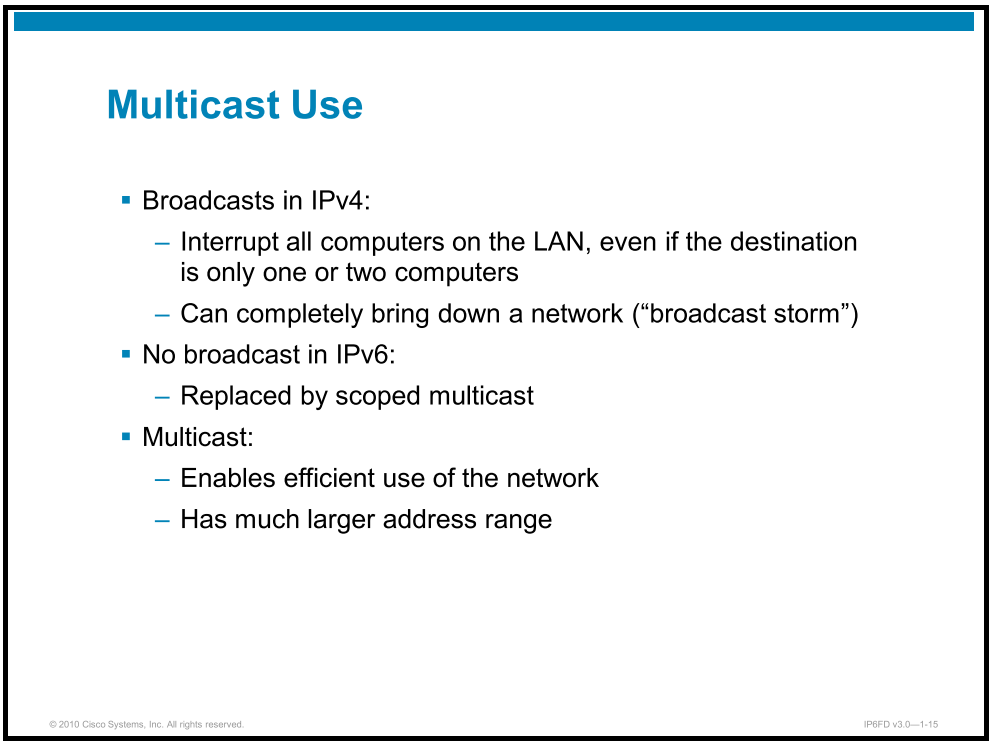


IPv6 has a much larger address space. This allows organizations to receive prefixes that are large enough for their present and future needs. Because of the strict aggregation of the global routing table, networks without portable prefix space (most small and mid-size organizations will have ISP-provided prefixes) must be renumbered when organizations change their upstream provider.

Renumbering is a very difficult, time-consuming, and error-prone task with IPv4. In IPv6, autoconfiguration enables simplified renumbering by sending the new prefix from the new upstream provider in the router announcements. Hosts will automatically receive the new prefix and then use the new address.

Note, however, that renumbering using this method does not take into account statically assigned systems, such as routers, switches, and servers registered in DNS. These nodes must still be renumbered by hand as they are with IPv4. IPv6, using autoconfiguration, can make the process of renumbering host devices easier.

Some operating systems use renumbering of the suffix as a security measure (for example, Microsoft Windows). Using the MAC address and EUI-64 suffix autogenerated from it, someone could track the host move around the IPv6 Internet. To prevent this, Windows uses randomization of the suffix to assure privacy.

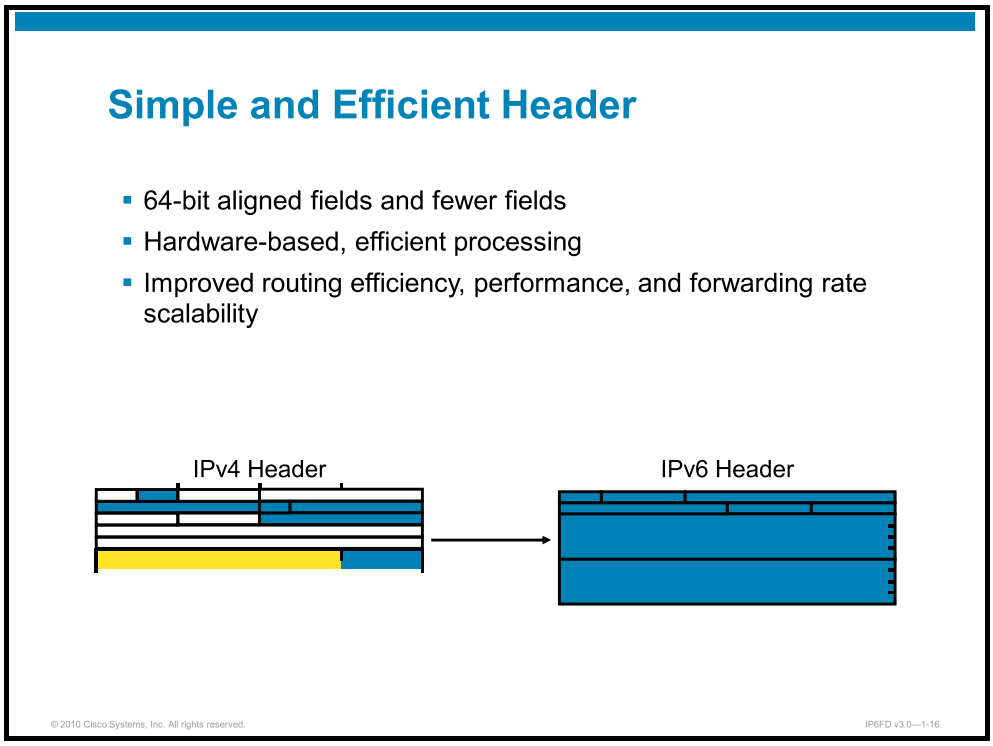


Broadcast in IPv4 causes many problems. It generates interrupts in every computer on the network, even if only one or two computers are involved. In some cases, broadcast completely brings down or severely degrades performance on a network; this is generally called a “broadcast storm.”

In IPv6, broadcast is replaced by multicast. Multicast enables the efficient use of the network by replacing most broadcast processes with more elegant multicast-based processes, using specific multicast groups for different functions. This prevents most problems caused by broadcast storms in IPv4. When an IPv4-style broadcast is needed, there is an all-nodes multicast address that is, essentially, a broadcast. IPv6 has no concept of “directed broadcast.”

The range of multicast addresses in IPv6, essentially 2 ^ 112, is much larger than in IPv4, so it should be easy to obtain a permanent multicast assignment for most services.

Multicast also has a scope parameter integrally defined in the address itself.



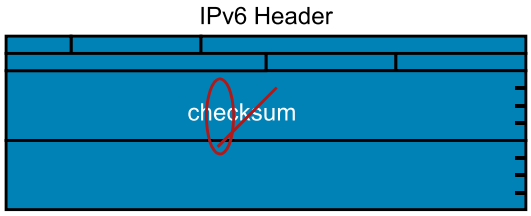
The new IPv6 header is simpler than the IPv4 header.

- Half of the previous IPv4 header fields are removed. This enables simpler processing of the packets, enhancing the performance and routing efficiency.
- All fields are aligned to 64 bits, which enables direct storage and access in memory by fast lookups.

These and other enhancements improve hardware-based processing, which provides scalability of the forwarding rate for the next generation of high-speed networks. This benefit, however, remains to be seen because 128-bit addresses are larger than the word-size of the current processors. This situation results in more lookup to obtain the complete 128-bit address.

No Checksum

- No checksum at the IP layer, no recalculation by the routers
- Improved routing efficiency, performance, and forwarding-rate scalability
- Error detection done by data link layer and transport layer

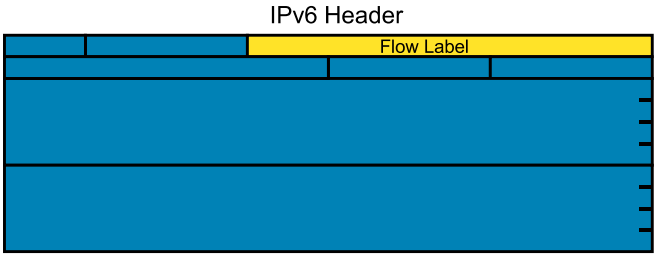


The IPv6 header is also simpler due to the removal of the checksum. Not only is the space in the header reused, but more importantly, the routers in the path do not do recalculation, which also provides routing efficiency.

This does not mean that there is no error detection—most data link layer technologies address error detection. Additionally, the transport layer that makes the end-to-end connection has a checksum that enables error detection. In IPv4, TCP checksums are available and User Datagram Protocol (UDP) checksums are optionally available. In IPv6, checksums are required for both transport protocols.

Flow Label Field Enables Per-Flow Processing

- A new flow label inside IP header
- Enables per-flow processing for differentiation at the IP layer
- Length of 20 bits

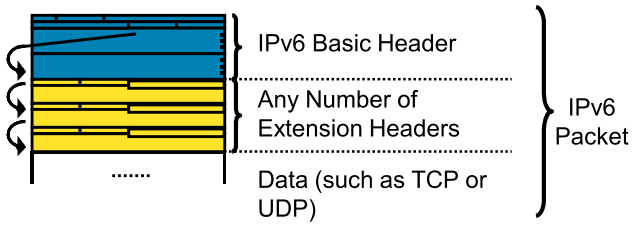


A new field has been added to the header in IPv6. The new Flow Label field enables per-flow processing by the routers in the path, which provides traffic differentiation at the IP layer without requiring additional work to identify the flows. With this label, a router does not have to open the transport layer segment to identify the flow; it finds the information in the IP packet header.

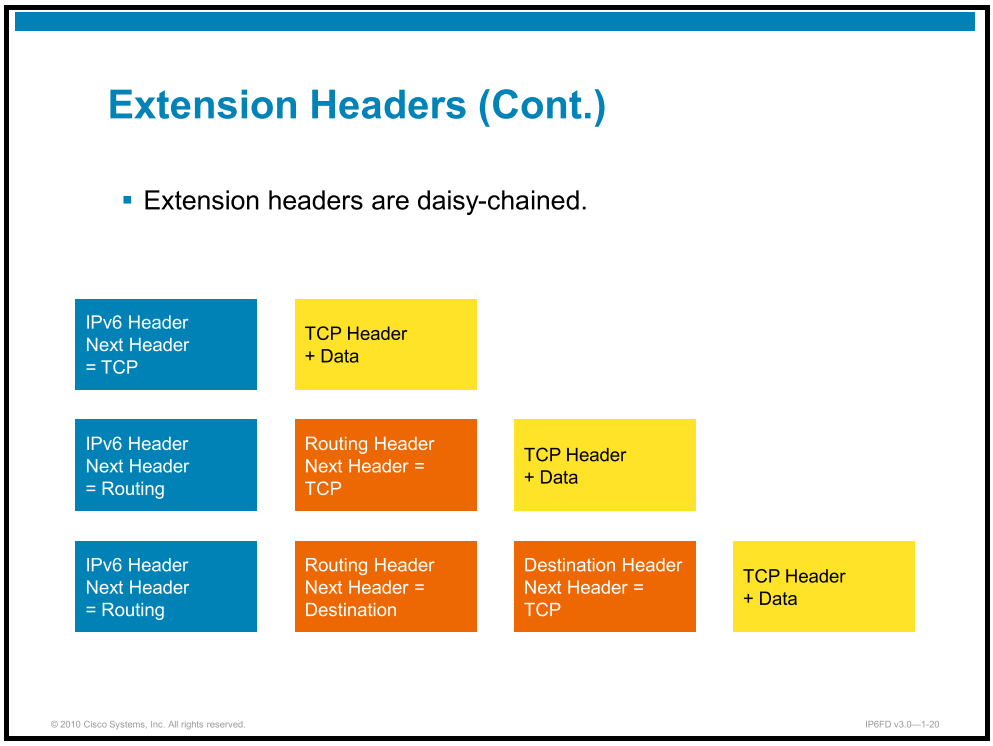
Note *IPv6 Flow Label Specification* (RFC 3697) specifies the Flow Label format and the requirements for IPv6 nodes labeling or forwarding flows. It does not, however, define a method of using the Flow Label to implement nondefault quality of service or any other services. This area remains a work in progress in the IETF.

Extension Headers

- Flexible extension headers
- More efficient handling of IP options
- Faster forwarding rate and end-node processing



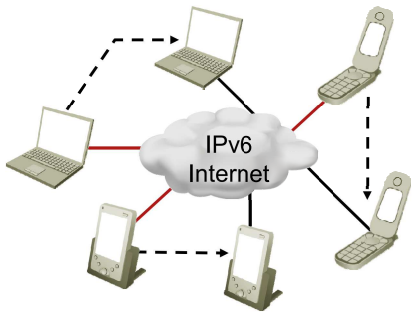
IPv6 uses a different approach from IPv4 to manage optional information in the header. It defines extension headers that form a chain of headers linked by the Next Header field contained in each extension header. This approach provides efficiency gains over IPv4 in the way that options and special functions are packaged. It enables a faster forwarding rate and leaves the router with less work to do for each packet.



Here is another view of extension headers. All extension headers are daisy-chained, each header pointing to the next header until they reach the transport layer data. This arrangement allows an IPv6 packet to be customized with features and functionality.

Mobility

- Mobile devices are fully supported while moving.
- Mobility is built in with IPv6—any node can use it.
- Efficient routing means performance for end users; elimination of “triangle routing” (using IP options) for return traffic as well.



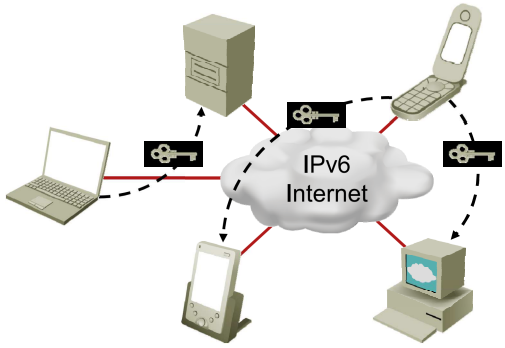
Mobility is a very important feature in networks of today. Mobile IP is an IETF standard available for both IPv4 and IPv6. Mobile IP allows mobile devices to change the IPv6 address of their current point-of-attachment without breaking existing Layer 4 sessions. In IPv6, mobility is likely to be widely supported, which means that most nodes will eventually be able to use it. In IPv4, mobility was not widely adopted, for a number of reasons.

The IPv6 routing header makes Mobile IPv6 much more efficient for end nodes than Mobile IPv4. IPv6 uses many enhancements. For example, Home Agent binding uses some header options (destination) that are mandatory for every IPv6 device. Route optimization is also supported in IPv6.

These optimizations eliminate “triangle routing” for incoming traffic to the endpoint. Using IPv6 header options, a direct path to the endpoint is communicated to the peer, which does not need to send the traffic to the home agent router anymore.

Security

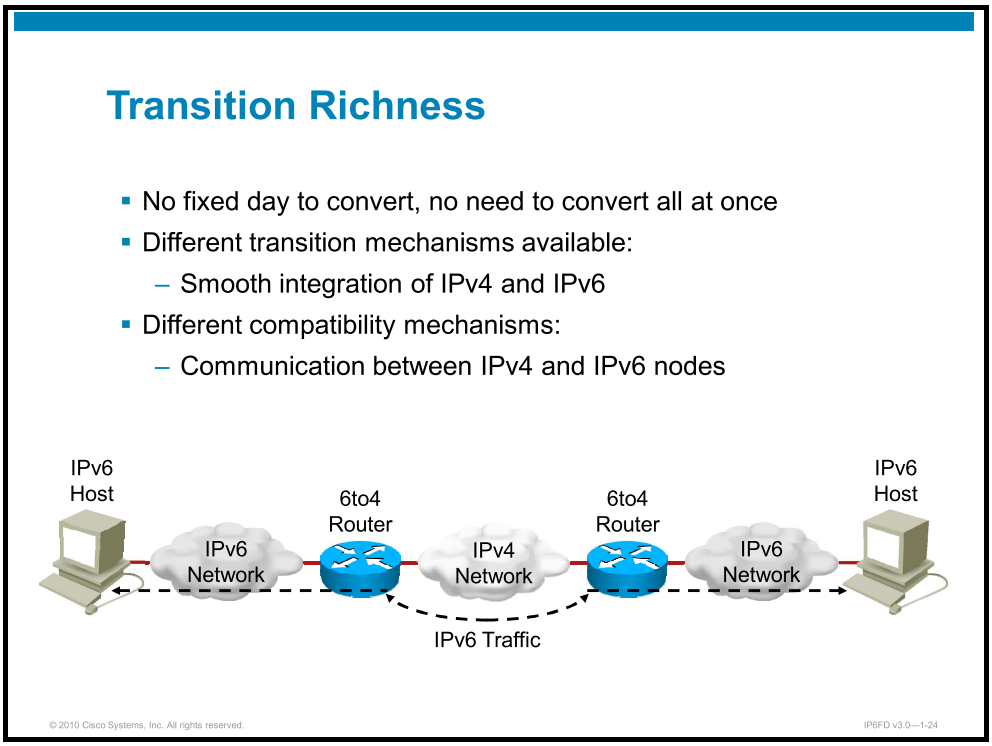
- End-to-end network security (integrity, authentication, confidentiality)
- Inherent (built-in) with IPv6—usable by any node



IP Security (IPsec) is the IETF standard for IP network security. IPsec ensures integrity, authentication, and confidentiality. IPsec is available for both IPv4 and IPv6, and is defined with the same RFCs. While the functionalities are essentially identical in both IPv4 and IPv6 environments, IPsec is mandatory in IPv6. This means that every IPv6 node will have IPsec enabled and will be able to use its features right away, allowing the IPv6 Internet to have better security because IPsec will be available on all nodes. IPsec also requires a key for each party, which, for wide deployment to be achieved, implies a global key deployment and distribution mechanism. This topic is beyond the scope of this course.

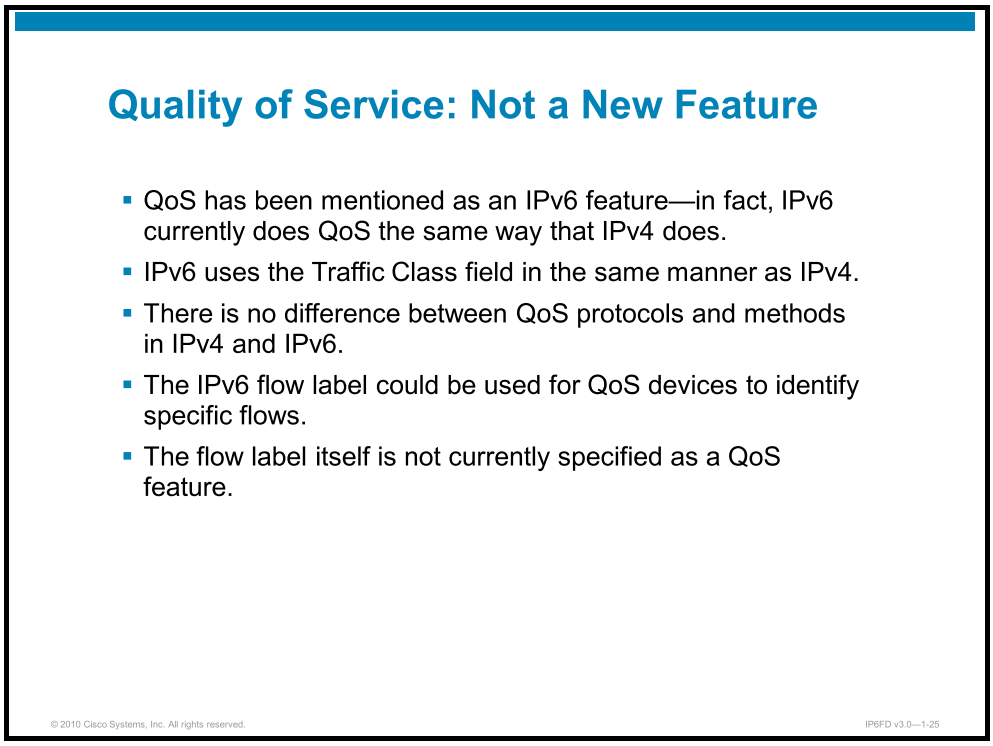
Transition Strategies to IPv6

This topic describes the transition strategies for IPv6.



The transition to IPv6 has been designed so that all nodes moving to IPv6 are not required to be upgraded at the same time. Many transition mechanisms have been designed to enable smooth integration of IPv4 and IPv6. Other mechanisms are available for compatibility, which IPv4 nodes can talk to IPv6 nodes, and vice versa. All these mechanisms can be applied to different situations and cases.

The figure shows one example of a transition and integration mechanism. The 6to4 routers automatically encapsulate the IPv6 traffic inside IPv4 packets. This mechanism is described in more detail in the “Describing IPv6 Tunneling Mechanisms” lesson.



QoS is often characterized as a new feature of IPv6. That is overstated. The current way to manage QoS in IPv4 is the same as in IPv6. The only difference is that IPv6 has a Flow Label field that can contain a label identifying a flow. This label is generated by the source of the packet. Having a flow label allows QoS devices in the path to make actions based on this label, but the existence of the label is not a feature of QoS.

However, the majority of QoS implementations still use the DiffServ approach (Per-Hop Behavior treatment of packets based on the local QoS policy).

IPv6 Technology Scope

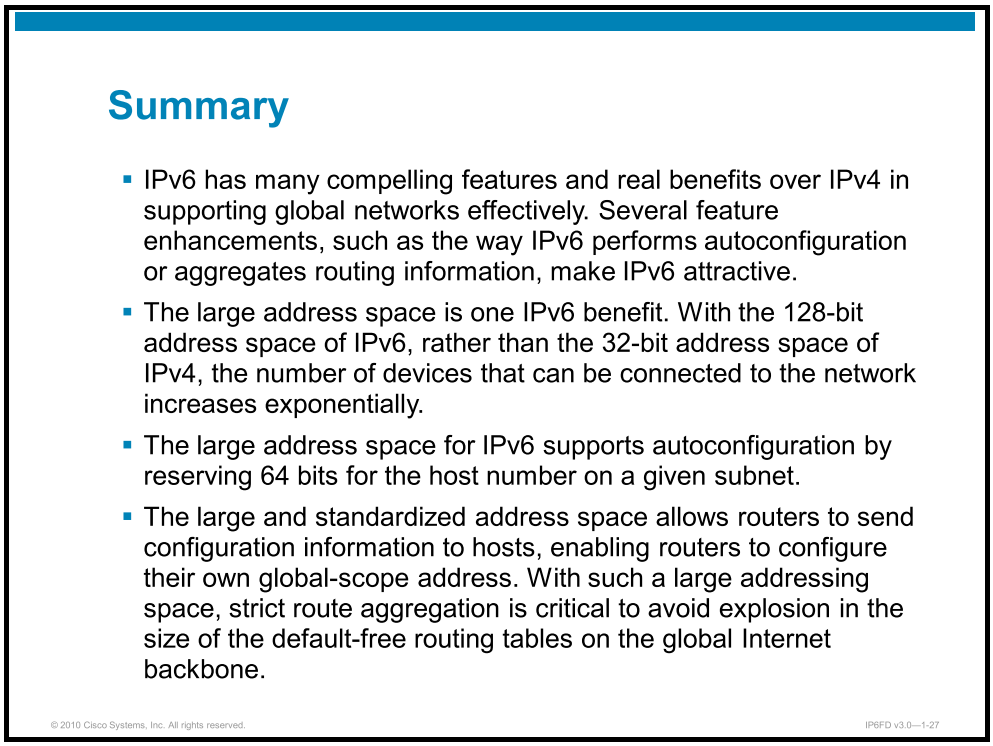
IP Service	IPv4 Solution	IPv6 Solution
Addressing Range	32-bit, Network Address Translation	128-bit, multiple scopes
Autoconfiguration	DHCP	Stateless, Stateful (DHCPv6)
Security	IPsec	IPsec-mandated, works end-to-end
Mobility	Mobile IP	Mobile IP with optimized routing
QoS	Differentiated service, integrated service	Differentiated service, integrated service
IP Multicast	IGMP, PIM, Multicast BGP	MLD, PIM, multicast BGP, scope identifier

IPv6 offers these general features and benefits:

- A large address space makes global reachability possible from every IPv6 node.
- Autoconfiguration is essential for deploying a large number of appliances. It would not be possible, practically, to manually configure IP addresses; you need some autoconfiguration mechanism that scales. DHCP may not be the right way to manage millions of clients.
- IPsec is mandated in the architecture.
- NAT includes end-to-end security in networks by requiring that you trust the end devices.
- Mobile IPv6 improves routing efficiency over IPv4.
- IPv6 is the same as IPv4 in QoS and header compression features. Both areas benefited from the work on IPv6. The IPv6 header compresses better than the IPv4 header because there are fewer fields.
- Other features are equivalent, except for a few details, such as scoped addresses in multicast, or the concept of stateless DHCP in which only static parameters are provided by the DHCP server.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 3

Understanding Market Drivers

Overview

This lesson describes the market drivers that help promote IPv6 as the key technology of the future. It defines the key markets for IPv6 solutions, outlines the importance of native IPv6 content and the relation to IPv6-only endpoints.

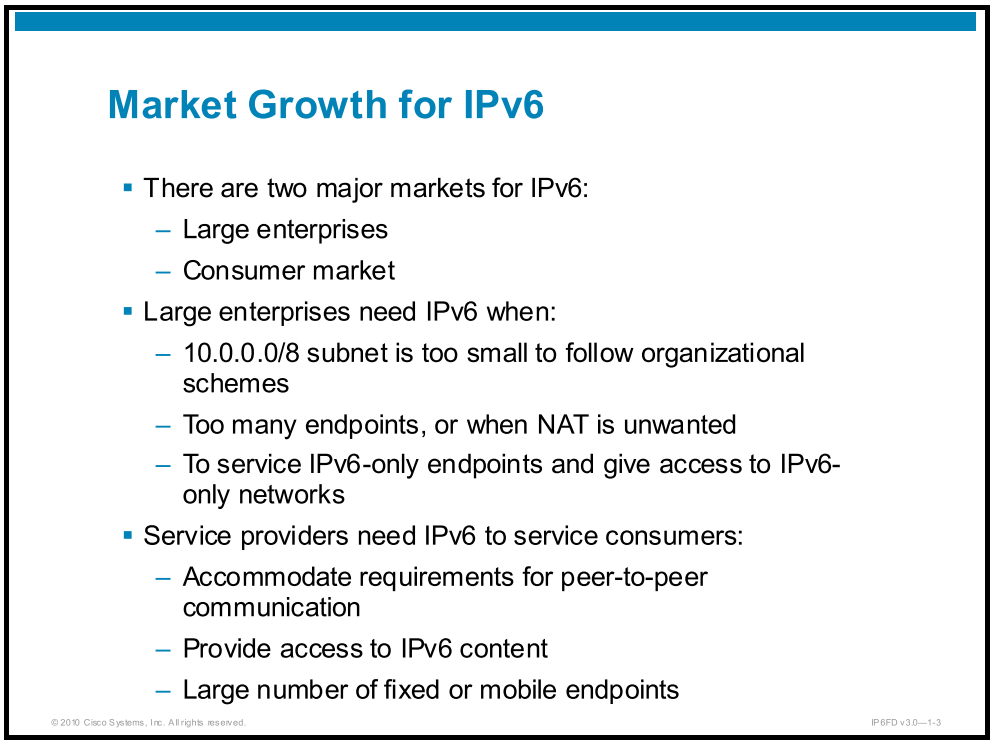
Objectives

Upon completing this lesson, you will be able to describe market drivers and the rationale to move to IPv6. This ability includes being able to meet these objectives:

- Describe the huge potential in market growth for IPv6 (mobile) endpoints
- Explain the importance of native IPv6 content and how its availability is a motivation to move
- Explain that IPv6 in Microsoft, open-source products, and government mandates are additional drivers for adoption

Market Growth for IPv6

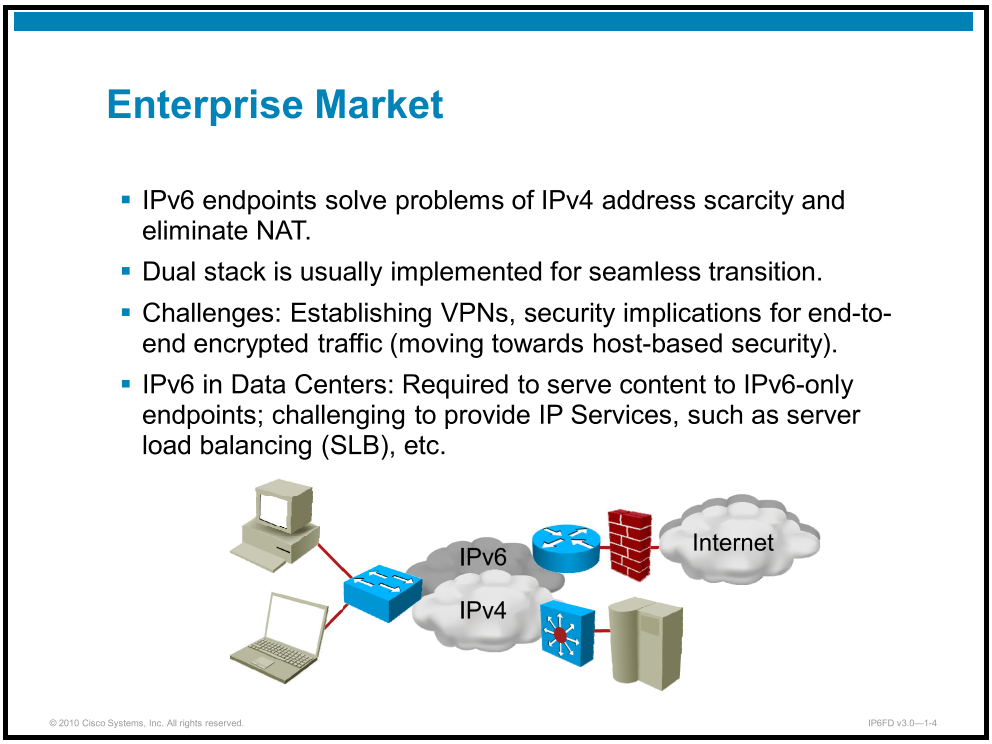
This topic describes the huge potential in market growth for IPv6 (mobile) endpoints.



The IPv6 protocol will experience usage growth in enterprise, service provider, and consumer markets. Enterprises and service providers have already been encouraged to adopt IPv6. The longer they postpone, the more it will impact them when their customers demand IPv6 services.

Large enterprises have already found themselves experiencing problems with IPv4 address scarcity in the private addressing blocks (10.0.0.0/8, 172.16.0.0–172.31.255.255/16, and 192.168.0.0–192.168.255.255/24). Either they can implement internal NAT and face the disadvantages of it blocking end-to-end communication, or they can move to IPv6. Many of the largest companies have chosen to adopt IPv6, but face problems with minor incompatibilities or features not yet available. Such examples are network management systems, IP Telephony, and proprietary applications. In the last five years, the majority of these issues have been resolved.

Service providers need to be prepared to provide their customers IPv6 services. Generally this does not require network redesigns or major hardware upgrades. The core technologies in service provider backbones remain MPLS and Multiprotocol BGP, which do support IPv6. Other applications, such as network management, still rely on IPv4, but progressively adopt IPv6.



Enterprise and Consumer Markets

The enterprise and consumer markets form the primary markets for IPv6 acceptance.

In the enterprise networks, IPv6 allows for end-to-end connectivity, meaning any server in any part of the network can accept connections without configuring address translations anywhere in the network.

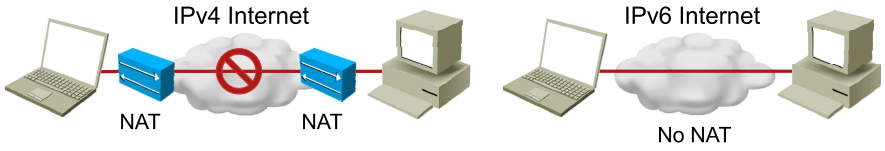
For seamless transition, dual stack is implemented in enterprise networks. Dual stack means that both IPv4 and IPv6 run concurrently on network devices (endpoints and routers) over the same Layer 2 network. IPv6 content and services are then accessed using the IPv6 bearer, and IPv4 content and services are accessed over the IPv4 bearer.

Enterprise customers require secure connections to establish virtual private networks (VPNs). VPNs secure communication flows over public networks, and this feature is built in into the IPv6 protocol stack natively. In enterprise networks, firewalls on network edges terminate such flows, decrypt them, and pass these flows unencrypted to secured parts of the network. However, security in IPv6 is meant to be end-to-end, and an encrypted flow “blinds” the firewalling device because the firewall does not terminate the secure connection anymore. Instead, the endpoint does; this introduces the necessity for host-based security, such as host-based firewalls and host intrusion prevention systems (HIPS).

Another aspect of IPv6 deployment is in data centers. When data centers start servicing IPv6-only endpoints, then IPv6 support in data center equipment will be required. The most challenging requirements are server load balancing (SLB) devices, proxies, and security devices. Network infrastructure supports IPv6 without major issues.

Consumer Market

- Adopts IPv6 where technically feasible and in areas with IPv4 address scarcity
- Many service providers invested in carrier-grade NAT (CGN) equipment, which currently solves the IPv4 address shortage problem but inhibits investments in IPv6
- End-to-end connectivity is excellent for peer-to-peer applications, such as file sharing and games
- IPv6 content availability is a prerequisite for growth!
  This relates to the IPv6 Enterprise/Data Center market segment



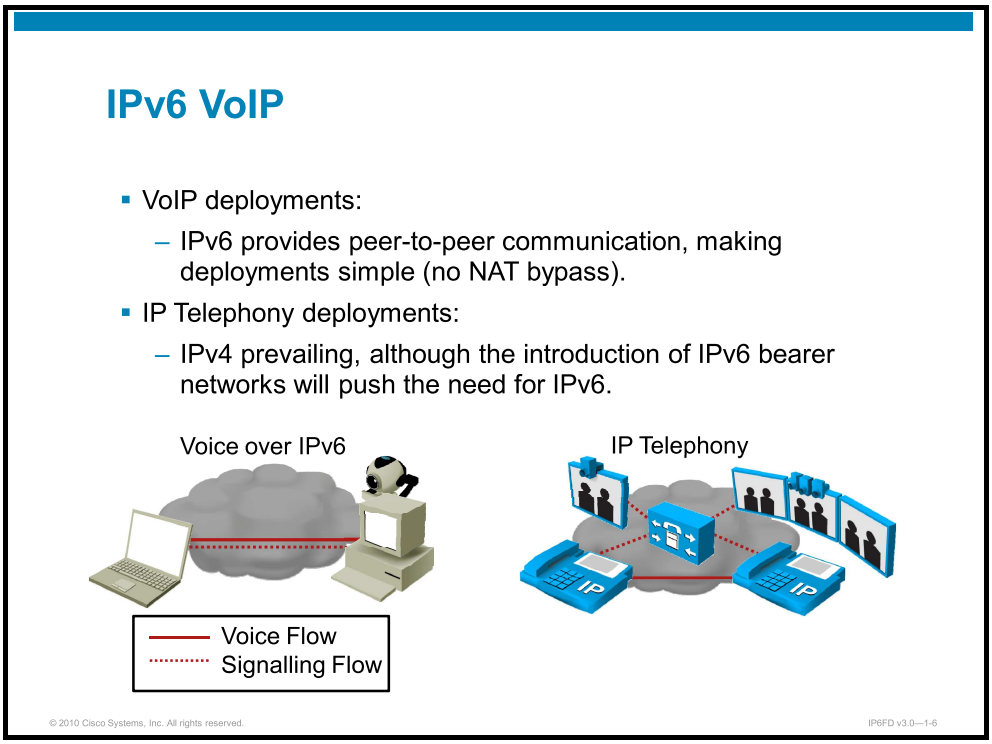
It is not really important to consumers whether they access information over IPv6, IPv4, or any other protocol, as long as they can obtain the information they desire. The goal of the network layer is to be transparent to higher communication layers.

If the user has an IPv6-only end device to access data, then the information must be served using IPv6. Using such end devices will contribute to the popularity of IPv6.

Currently, many service providers (and mobile operators) have invested in IPv4 solutions. These solutions include carrier-grade NAT (CGN) or A+P, which introduce additional translation points or assign only a number of ports (Layer 4, in the range of ~200) to a user. Compare this solution to the unlimited ports assigned in classic implementations. Typically, service providers introduce CNG and IPv6 in parallel.

However, the absence of NAT in IPv6 brings substantial advantages to the end consumer, especially when using peer-to-peer applications such as file sharing, VoIP (Skype and others), and online games. Such systems can connect directly without having to use proxies or find creative ways to bypass NAT and firewalls.

IPv6 content availability is something like a “chicken and egg” problem—which one comes first? Should IPv6-only endpoints be deployed first to influence the requirement for content served on IPv6? Or, should data centers and IP Services upgrade to IPv6 to be able to serve IPv6 content first?



Common Technologies

The common technologies for adopting IPv6 are VoIP, IP Television, IT business support, and education.

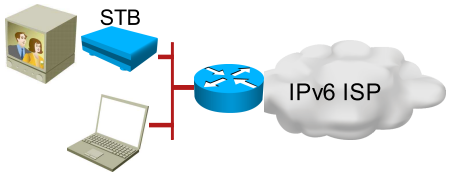
The IPv6 addresses peer-to-peer communication and enables it to work end-to-end, that is, without NAT or firewall bypassing techniques. This will boost the use of IPv6 for VoIP applications that follow the peer-to-peer model. The voice data stream can be transferred directly between the endpoints, improving connection setup speeds, reducing the amount of signaling, and so on.

On the contrary, deployment of IP Telephony systems—enterprise IP Telephony solutions such as Cisco Unified Communications Manager, Cisco IP phones, and so on—are bound to networks of smaller and manageable size, which generally IPv4 can accommodate. For truly global enterprises, IPv6 can simplify IP Telephony networks. IPv6 will become mandatory when IPv4 is removed in at least one part of the enterprise network.

Note Peer-to-peer communication is otherwise praised in the consumer market because it allows file-sharing applications and games to function without obstacles (such as NAT). Such traffic does consume service provider bandwidth and resources, but on the other hand, Internet access brings revenue to service providers.

IP Television

- IPTV uses multicast data plane and control plane: IPv4 multicast, IGMP, SSM/PIM.
- The data is usually transmitted into a VPN—a network separated from other data networks to provide stability and QoS; typically IPv4-based.
- IPv6 has services to accommodate multicast data transmission and multicast routing.
- IPTV will switch to IPv6 when the bearer will be IPv6-only; CPE IPv6 support is crucial to achieve this goal.

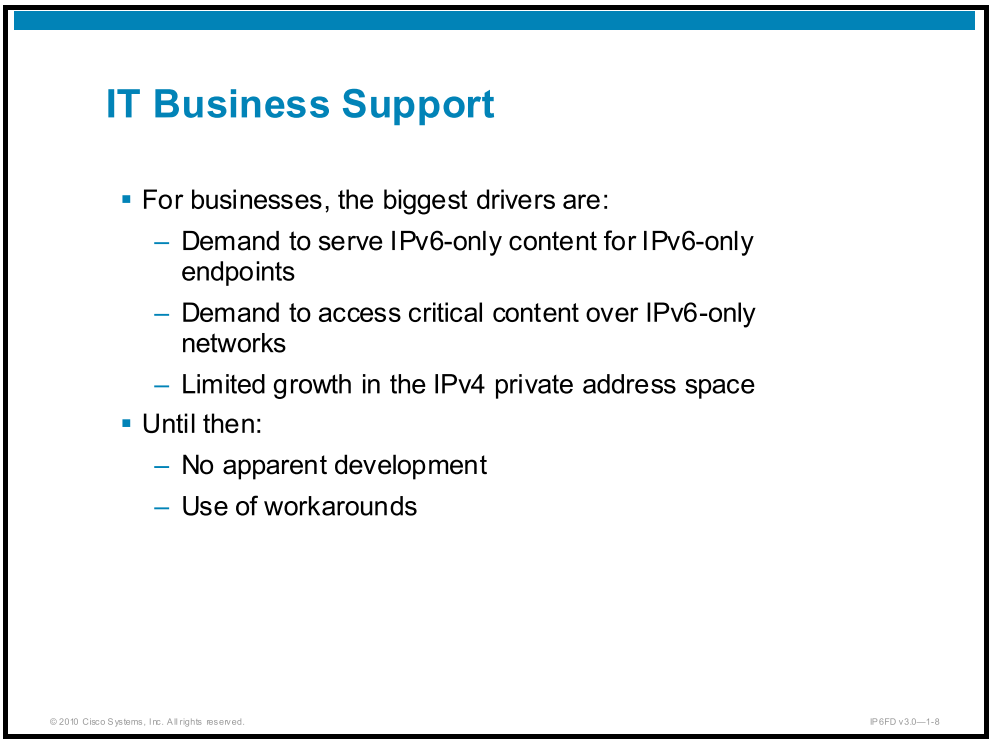


IPTV deployments are, similarly to IP telephony deployments, bound to closed networks at the service provider. The customer base at the service provider defines the size of the network, which can currently be accommodated using IPv4 solutions.

Service providers introduce separate networks to carry IPTV. By using this approach, IPTV traffic is not mixed with other customer traffic, and it is, therefore, possible to maintain screen image and sound quality.

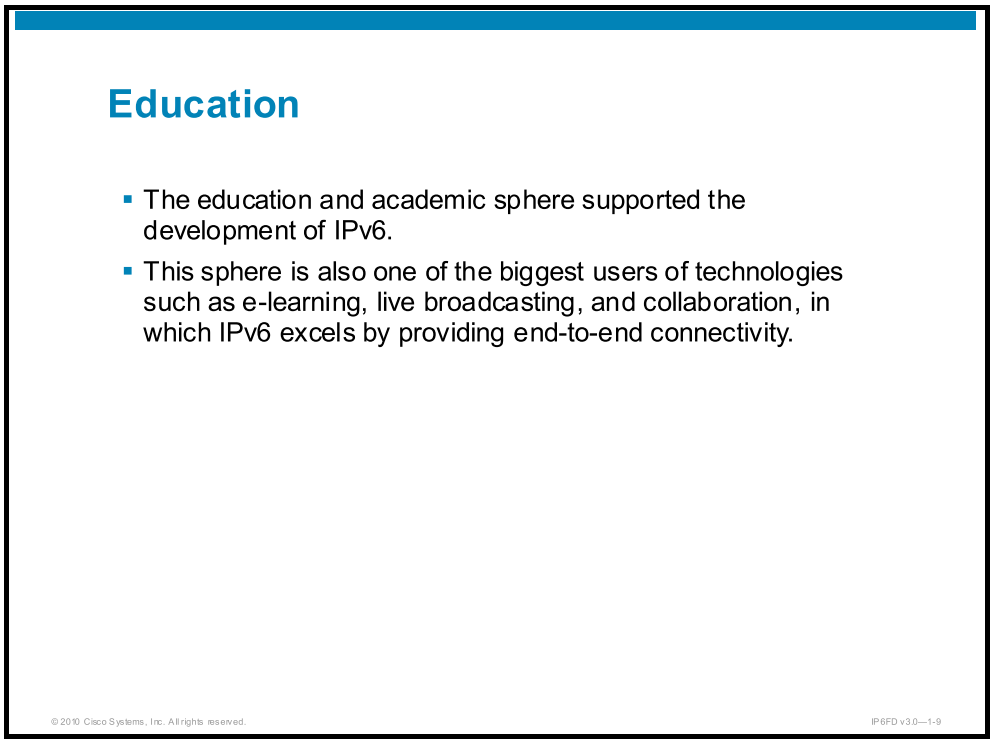
A driver to move IPTV to IPv6 would be a shutdown of IPv4 services and easier IPv6 multicast deployment in contrast to IPv4. More multicast groups are available in IPv6, and addressing is easier to design. A move to an IPv6-only user access would be feasible, in this case, and will encourage IPv6 development for missing components.

Implementing IPv6 on the customer premises equipment (CPE) is not particularly challenging, but on the other side, service providers are trying to keep CPE costs as low as possible. Since users demand IPTV CPE to support recording on built-in disk drives, and so on, IPv6 support should not be a big issue.



When implementing IPv6, enterprise clients often encounter problems with equipment compatibility or software support. Such examples are management software, security products, authentication servers, server load-balancing devices, firewalls in failover configurations, and so forth. This does not stop the deployment of IPv6, but prolongs the dual-stack setup in which mixed IPv4 and IPv6 services are deployed.

Obviously, the demand for critical content from the client side will push the development of IT business services to move to IPv6. Several business applications—from enterprise resource planning to customer relationship management, desktop virtualization, and so on— will benefit from moving to IPv6. The main drivers are system scalability and logical network simplicity; both of these decrease the cost of owning a network.



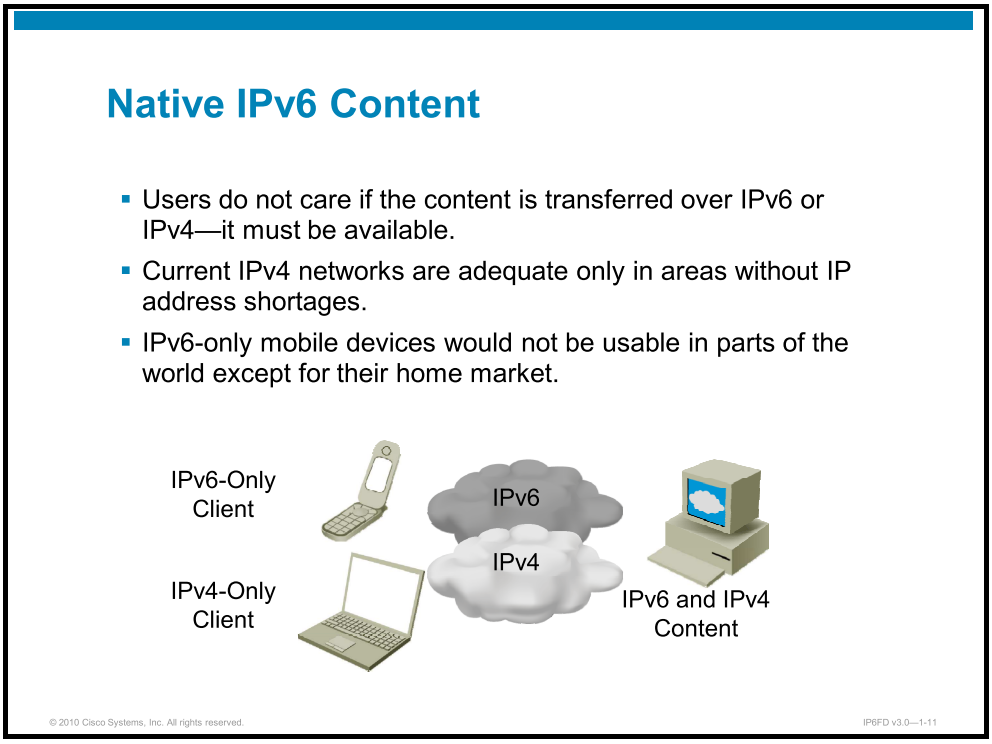
The academic education sphere has been adopting IPv6 since its beginnings and has pioneered many of the technologies that businesses use today.

For business education services, the benefit of IPv6 will be similar to the benefits of using peer-to-peer VoIP, videoconferencing, collaboration, and IPTV. These solutions use either point-to-point connectivity, where IPv6 excels, and they use the benefit of good scalability and simplicity. IPv6 e-learning solutions will require less time to set up, and they will use bandwidth more efficiently by taking advantage of IPv6 QoS and multicast routing capabilities.

Collaboration performance requirements can be compared to the requirements for online gaming; the goal is efficient connectivity between several users and sites.

Native IPv6 Content

This topic describes the importance of native IPv6 content and how its availability is a motivation to move.



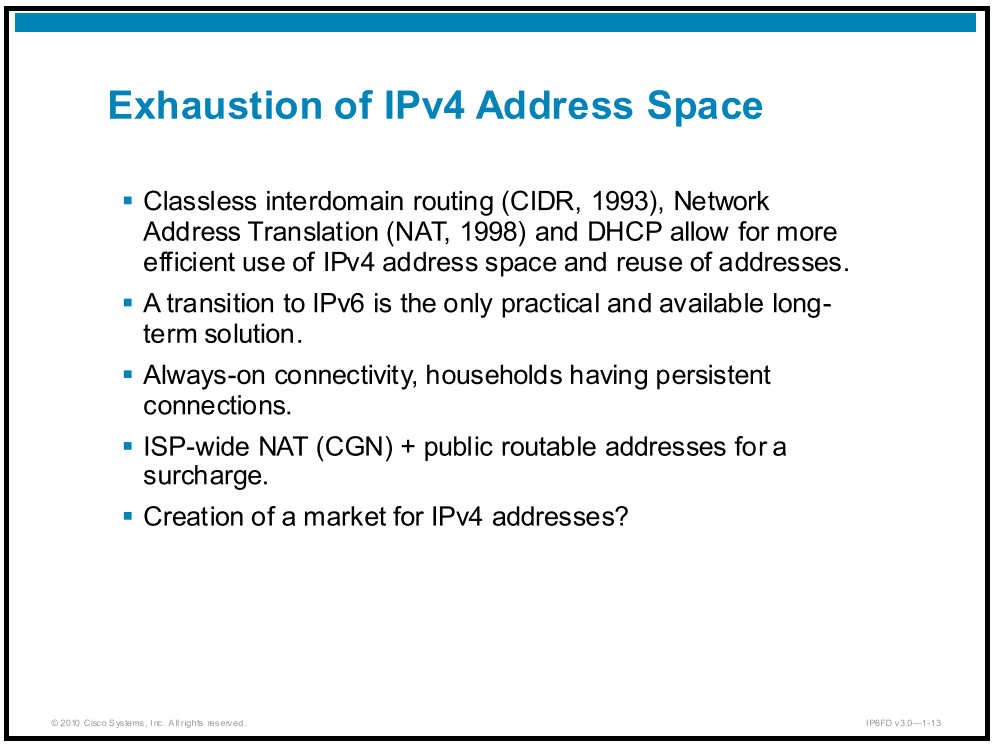
The growth of IPv6 devices in markets with IPv4 address shortages will have a worldwide impact. The most affected market is the Asian region. Because Asian countries are the biggest producers of electronics, it is expected that new (mobile) devices will come with a native IPv6 protocol stack. However, these devices will lack the IPv4 stack due to its limitations and additional implementation costs.

Traditional IPv4 networks will need immediate upgrades to support such IPv6-only endpoints if the content providers will not want to lose revenue from these IPv6-only users.

On the other hand, all content providers will switch to dual-stack configurations, serving the same content both over IPv4 and IPv6. This will remain for some time (maybe decades) to support IPv4-only devices in the IPv6 next-generation Internet.

Drivers for Adoption

This topic describes how IPv6 in Microsoft, open-source products, and government mandates are additional drivers for adoption.



Exhaustion of IPv4 Addresses

This subtopic outlines the problem of IPv4 address exhaustion and points to the solutions for addressing it.

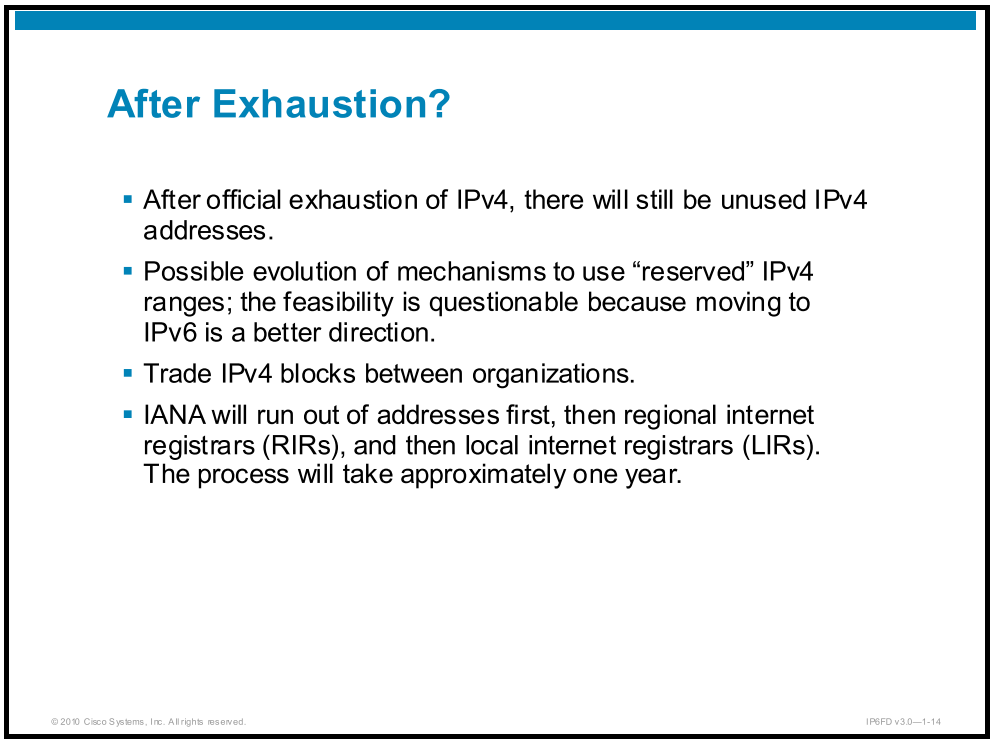
There have been several enhancements in the IPv4 Internet to improve the efficiency of how the address space is used. The distribution of IPv4 addresses within classes (especially in larger classes) is scattered, and the host density ratio is usually poor.

In certain markets, service providers attracted customers by offering them always-on connectivity. For this connectivity, the service provider needed a pool of IP addresses that, once assigned, could not be used for another user because these addresses were constantly "occupied." Users grew accustomed to it and did not want to settle for anything less, which then left fewer unused addresses available.

A small number of service providers introduced carrier-grade NAT (CGN) devices that performed ISP-wide NAT. Consequently, customers were unable to accept inbound connections from the Internet. For customers that need to accept incoming connections, this option was usually offered with the possibility of obtaining a complete public routable IPv4 address for a surcharge.

CGN is now used mostly by mobile operators for their users that access the Internet using mobile phones, which usually do not host IP Services. Thus they do not need incoming connections. Mobile third-generation networks are asymmetrical in terms of user-available bandwidth (upstream much less than downstream to the user). Therefore, hosting services do not make sense on a mobile device, and global inbound reachability is not needed.

One idea is to make address blocks available for sale, but that would conflict with the definition that IP addresses are free. Secondly, once bought, they would not be resold without reasonable profit. Thirdly, buying an IPv4 block and renumbering your hosts every time an enterprise grows costs more than switching to IPv6 once.

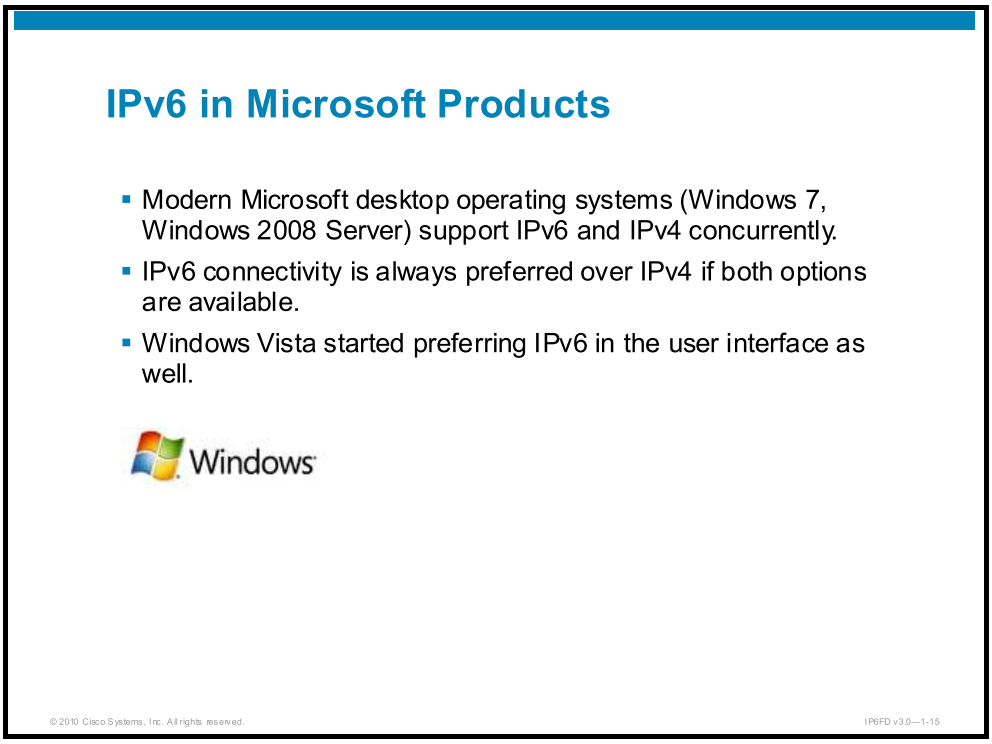


After official IPv4 address exhaustion, there will still be IPv4 addresses available, but only at organizations. Trade for this space among organizations could be possible, but introduces legal issues and additional complexity to the global routing table.

Another option would be to use the reserved IPv4 ranges, such as Class E (240.0.0.0–255.255.255.254), but this effort would require software updates at both host and network devices and has been rejected by IETF. The necessary effort and cost would be comparable or worse than switching to IPv6 entirely.

One of the most »famous« recent recoveries was the reclamation of the 14.0.0.0/8 IPv4 address range with assignments from 15 years ago. This range was once used to interconnect the Internet with other networks.

Forecasts are that IANA will run out of IPv4 addressing space in 2011–2012, and consequently, regional and local Internet registrars will run out of allocations after that. It is possible that this process might advance more slowly as IPv6 adoption in most “crowded” markets grows.



IPv6 in Modern Operating Systems

This topic describes IPv6 support in the most popular operating systems, including Microsoft and open-source platforms.

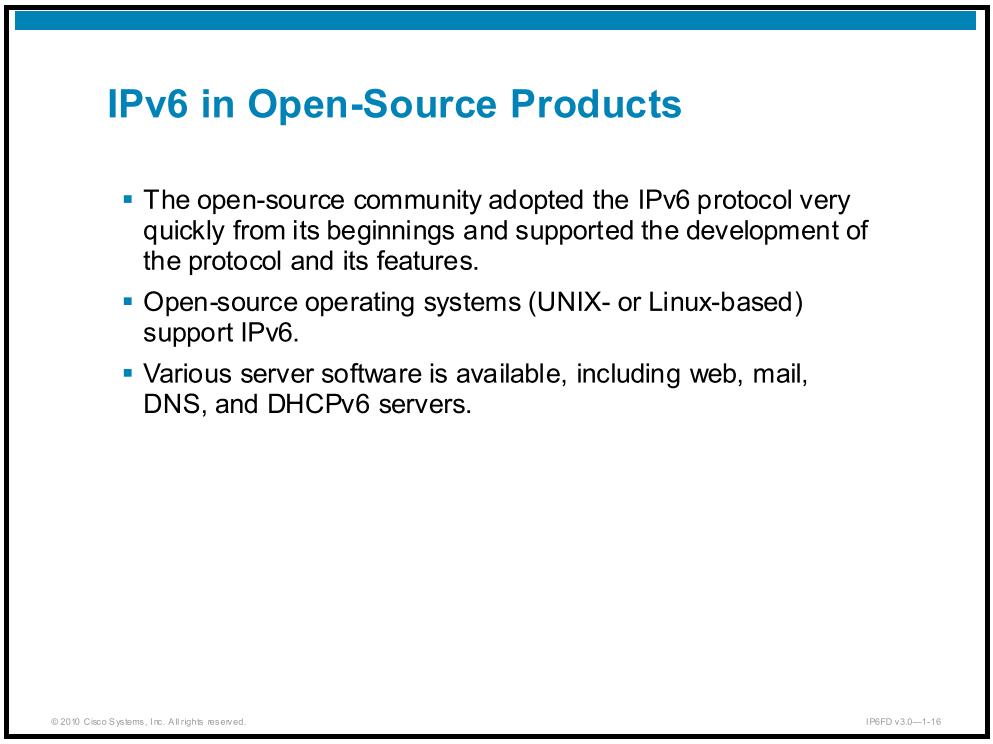
The new Microsoft Windows[1] operating systems support IPv6 natively. Windows XP already had an IPv6 protocol stack, but it was managed through the command line. In previous versions of Windows, the IPv6 protocol stack had to be added manually.

IPv6 connectivity is preferred over IPv4 if both options are present. This is one of the ways to promote the use of IPv6, but it brings “discomfort” as well: if the IPv6 path is not functional in the enterprise network, the user will not have connectivity unless the application or TCP stack falls back correctly to IPv4.

The operating system will find an operating IPv6 stack in the system, and it will default to it. However, it cannot detect if the upstream network is not fully functional, and the traffic will be lost. The user has two options: either disable the local IPv6 stack or make an IPv6 path functional towards the ISP.

The support of IPv6 in Microsoft Windows is of key importance in enabling user access to IPv6-based content. Network infrastructure and IP Services (such as DNS, and so on) must be enabled.

[1] Windows logo: source Microsoft



The open-source community contributed to the development of the IPv6 protocol suite. This explains why IPv6 is very well supported in operating systems based on open-source code.

Major credit also goes to the academic communities and research networks, especially to the community maintaining the IPv6 test ground—the “6bone” network. This network ceased operation when it achieved its goals: IPv6 standards development and testing.

For UNIX and Linux operating systems, all sorts of server software is available: from web servers, to email servers, DNS servers, DHCPv6 servers, and others.

IPv6 on Mobile Platforms

- Mobile devices are the key demand generators for IPv6 networks.
- Large potential base of users.
- Large areas of the world without “wired” network coverage can be covered with wireless networks (3G and 4G, WiMax, etc.).
- One person can have many mobile devices and in-house devices that need to synchronize online.

IPv6 on Other Platforms

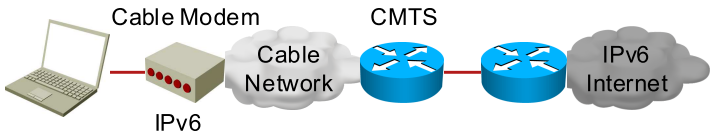
Here we list important platforms that will encourage the development and deployment of IPv6: mobile platforms, cable networks, and industrial use.

Mobile users are one of the first target markets for IPv6 deployments because, in many countries, the wired infrastructure is not able to allow multiple millions of users. Wireless infrastructure (with mobile clients) will be used instead. User additions generally do not require upgrades to the infrastructure, and if upgrades are necessary, a large number of users will benefit. The crucial markets for IPv6 development in this area are Asian countries with high population density and large potential for mobile network growth.

IPv6 will also accommodate the fact that one person can have many devices with IP addresses, including mobile phones, mobile email appliances, home theater equipment, and devices for intelligent home systems. These devices need to communicate with each other and be online at the same time, each one requiring an IP address. With market growth, the IPv4 address shortage problem becomes even worse.

IPv6 on Cable Networks

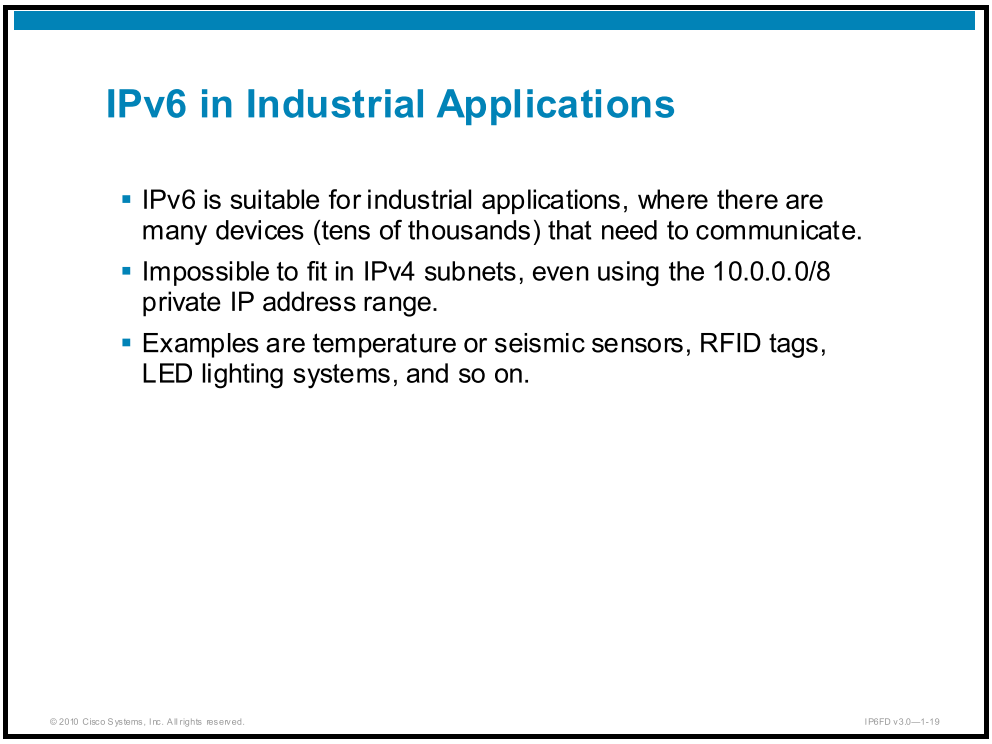
- Every cable modem needs an IP address.
- Large cable service providers have run out of IPv4 address space to manage cable modems and have moved to IPv6.
- IPv6 addresses are supported on DOCSIS 3.0 devices.



Large cable operators are adopting IPv6 as well. In the cable modem world, every cable modem—as a customer premises equipment (CPE) device—needs an IP address to communicate. Obviously, this address must come out of a range that a service provider uses, and it must be publicly reachable for the user to have access to the Internet.

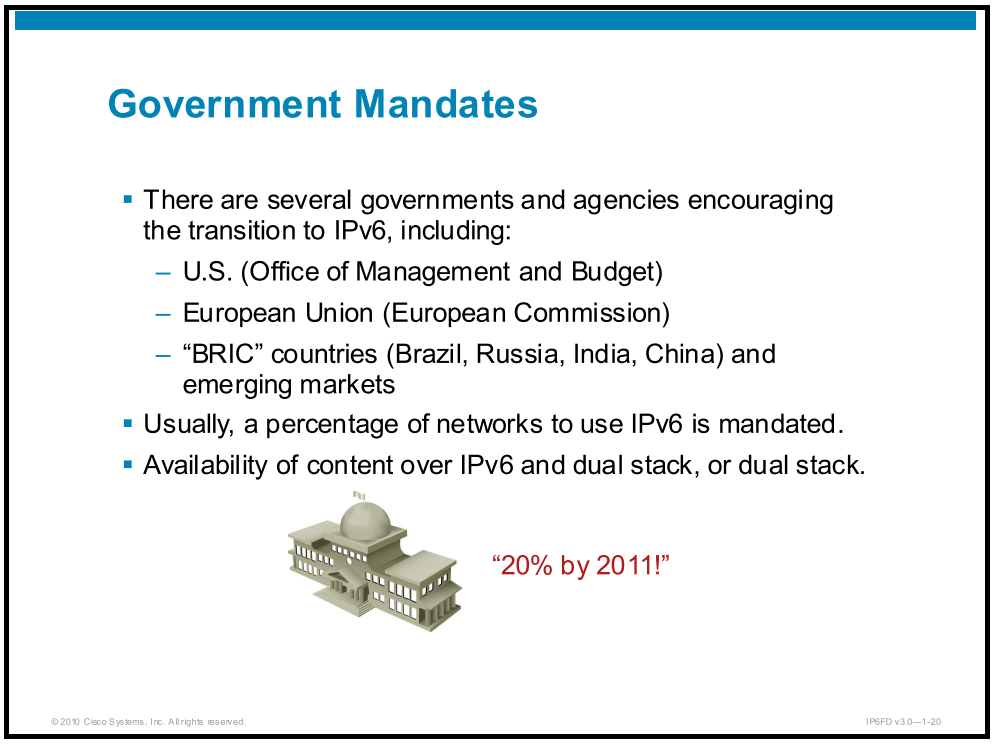
IPv6 fits in very large cable networks, in which service providers need to manage a large number of cable modems inside a single system. This scenario cannot be achieved using IPv4 as it will reach its limits of scalability. Major cable providers are performing tests to switch to IPv6.

The communication in cable networks is defined using the Data-over-Cable Service Interface Specifications (DOCSIS) standards, and the latest version that allows IPv6 addresses is DOCSIS 3.0. IPv6 was also supported as an interim solution on devices adhering to “DOCSIS 2.0 + IPv6” specifications.



In industrial environments, IPv6 is usable in micro-devices that perform a specific task and return the result to a central management system. Examples of such devices are sensors, valves, lighting objects, electrical measurement devices (Smart Grids) etc., which can be controlled remotely. For remote control, however, they need to be reachable. With IPv6, it is possible to deploy thousands of devices—each one with its own IPv6 address—and to configure them using IPv6 autoconfiguration methods.

Note LED screens on the 2008 Beijing Olympics were controlled by LED arrays, each one having its own IPv6 address.



Government Mandates

Governments around the globe encourage the adoption of IPv6 to maintain competitiveness of their economies.

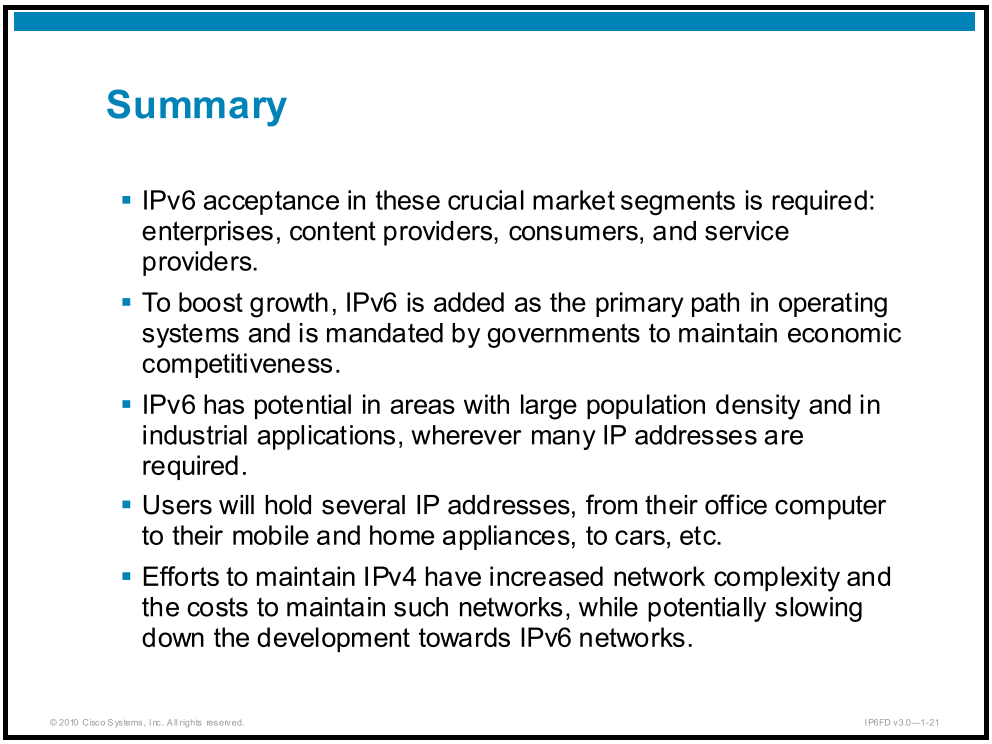
Insisting on IPv4 networks and developing mechanisms to increase the utilization of IPv4 addresses adds complexity to networks. These complex networks need maintenance, which involves higher operating costs. These costs could be avoided if the networks developed towards simpler protocols, such as IPv6, which are more scalable and easier to maintain.

To boost the efficiency in enterprises and increase competitiveness economy as a whole, governments recognized the opportunity and mandated the use of IPv6 in networks. Such examples are, among others, the European Union (with the European Commission to enforce the rules) and the U.S. (Office of Management and Budget).

The rules define the amount of content to be accessible through IPv6 and the percentage (or portion) of public entity networks to use IPv6 as their core protocol, or dual stack.

Summary

This topic summarizes the key points that were discussed in this lesson.



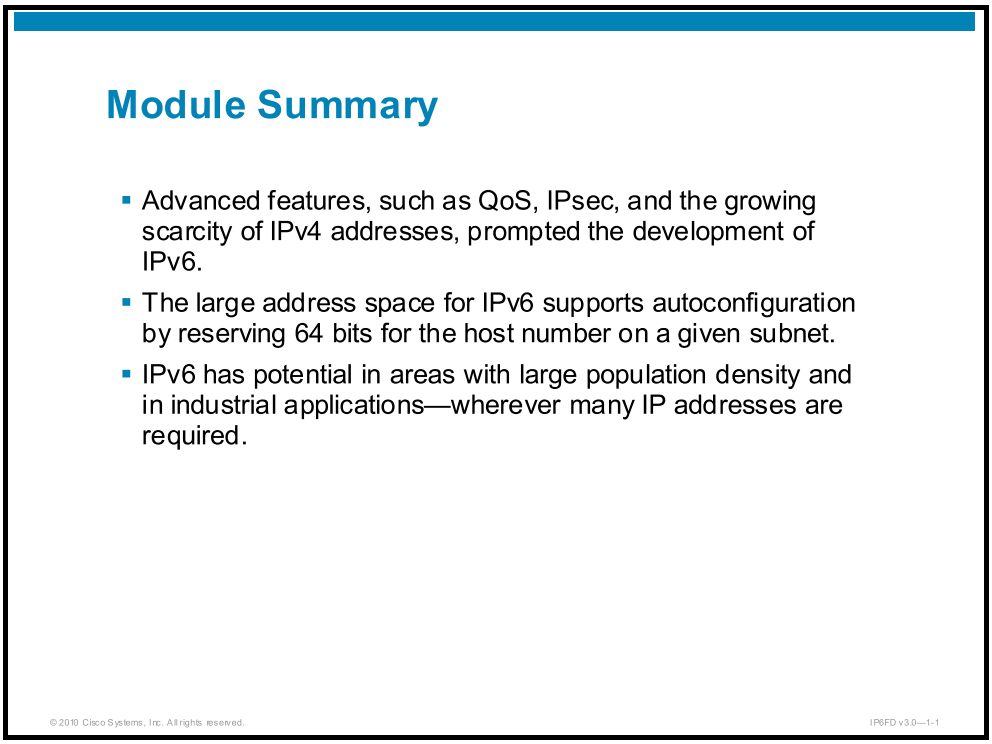
References

For additional information, refer to these resources:

- A Pragmatic Report on IPv4 Address Space Consumption
  http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html
- Microsoft Internet Protocol Version 6 (IPv6)
  http://technet.microsoft.com/en-us/network/bb530961.aspx
- US Government using IPv6
  http://ipv6.com/articles/general/US_Government_IPv6.htm
- European Commission IPv6 Task Force
  http://www.eu.ipv6tf.org/in/i-index.php#
- CableLabs Issues DOCSIS 3.0 Specifications Enabling 160 Mbps
  http://www.cablelabs.com/news/pr/2006/06_pr_docsis30_080706.html
- IPv4 Address Exhaustion
  http://en.wikipedia.org/wiki/IPv4_address_exhaustion#Mobile_devices

Module Summary

This topic summarizes the key points that were discussed in this module.



Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) Which three are IPv4 address conservation mechanisms? (Choose three.) (Source: Explaining the Rationale for IPv6)

A) NAT
B) CIDR
C) address subnetting
D) address allocation
E) DHCP

Q2) New feature support, such as quality of service, IP security, and mobile IP, were driven by what force? (Source: Explaining the Rationale for IPv6)

A) DARPA
B) ARPA
C) address shortages
D) Internet growth
E) defense contracts

Q3) Which two issues did NAT create while helping with IPv4 address conservation? (Choose two.) (Source: Explaining the Rationale for IPv6)

A) breaks the end-to-end model
B) requires no tracking of translation transactions
C) requires end-to-end security
D) breaks end-to-end security
E) requires the deployment of application-layer gateways

Q4) The IPv6 header is constructed differently from an IPv4 header. Which two key benefits result from the redesign? (Choose two.) (Source: Evaluating IPv6 Features and Benefits)

A) increases the number of lookups for processing
B) aligns the header on 64 bits to allow for faster processing
C) reduces the number of lookups for processing
D) removes a number of fields, thus requiring less processing
E) changes the header placements, requiring less processing

Q5) In which way will security in IPv6-enabled environments be different? (Source: Evaluating IPv6 Features and Benefits)

A) IPsec support is mandated for every IPv6 node.
B) IPsec has been improved for IPv6 nodes.
C) IPsec support in IPv6 now includes a global key deployment.
D) IPsec is no longer used in IPv6 networks.

Q6) Which three IPv4 header fields were dropped when IPv6 was created? (Choose three.) (Source: Understanding Market Drivers)

A) IHL
B) Type of Service
C) Header Checksum
D) Flags
E) Flow Label

Q7) The extension headers serve which important function in IPv6 networks? (Source: Understanding Market Drivers)

A) identify optional processes that can be run on each IPv6 packet
B) allow IPv6 nodes to manipulate routers
C) identify processes that manipulate the routers in the path of a packet
D) replace the traditional role of TCP and UDP in a network

Module Self-Check Answer Key

Q1) A, B, E

Q2) D

Q3) A, D

Q4) B, D

Q5) A

Q6) A, C, D

Q7) A

Module 2

IPv6 Operations

Overview

As a Layer 3 protocol, IP version 6 (IPv6) has a broad impact on the operations of the network and the interactions with other systems and other layers of the Open Systems Interconnection (OSI) model. This module describes the structure of the IPv6 address format, how IPv6 interacts with data link layer technologies, and how IPv6 is supported in Cisco IOS Software. With an understanding of basic IPv6 operations, you will be more successful in your IPv6 integration efforts.

Module Objectives

Upon completing this module, you will be able to describe the structure of the IPv6 address format, how IPv6 interacts with data link layer technologies, and how IPv6 is supported in Cisco IOS Software. This ability includes being able to meet these objectives:

- Describe the IPv6 addressing architecture, including types of addresses and address representation
- Describe changes in the IPv6 header and the purpose of extension headers
- Configure IPv6 on the Windows XP, Windows 7, and Linux operating systems
- Describe and use Cisco IOS commands to enable IPv6 on Cisco routers
- Describe ICMPv6 message types and how they are used to troubleshoot IPv6 issues, and describe the neighbor discovery protocol
- Configure and troubleshoot a Cisco IOS router to support IPv6 operation

Lesson 1

Understanding the IPv6 Addressing Architecture

Overview

The IP addressing architecture is fundamental to how addresses are allocated, how nodes are numbered, how routing tables are constructed, and how packets are routed throughout the network. With a 128-bit address length, the IP version 6 (IPv6) address space is significantly larger and more diverse, and thus is more complicated to manage. This lesson describes the IPv6 addressing architecture, including types of addresses and address representation and the types of addresses that different types of devices are expected to have.

Objectives

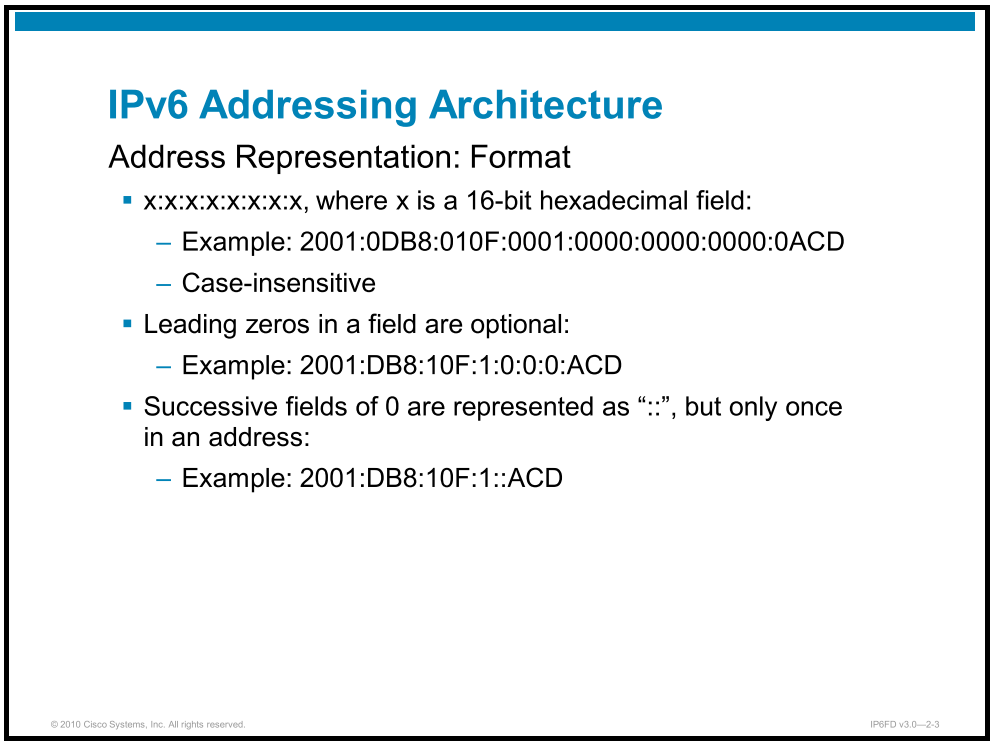
Upon completing this lesson, you will be able to describe the IPv6 addressing architecture, including types of addresses and address representation. This ability includes being able to meet these objectives:

- Describe the IPv6 addressing architecture
- Describe the format and uses of the various types of IPv6 addresses
- Create and use the various types of IPv6 addresses

Determine the required IPv6 addresses for an IPv6 host and an IPv6 router

IPv6 Addressing Architecture

This topic describes the IPv6 addressing architecture.

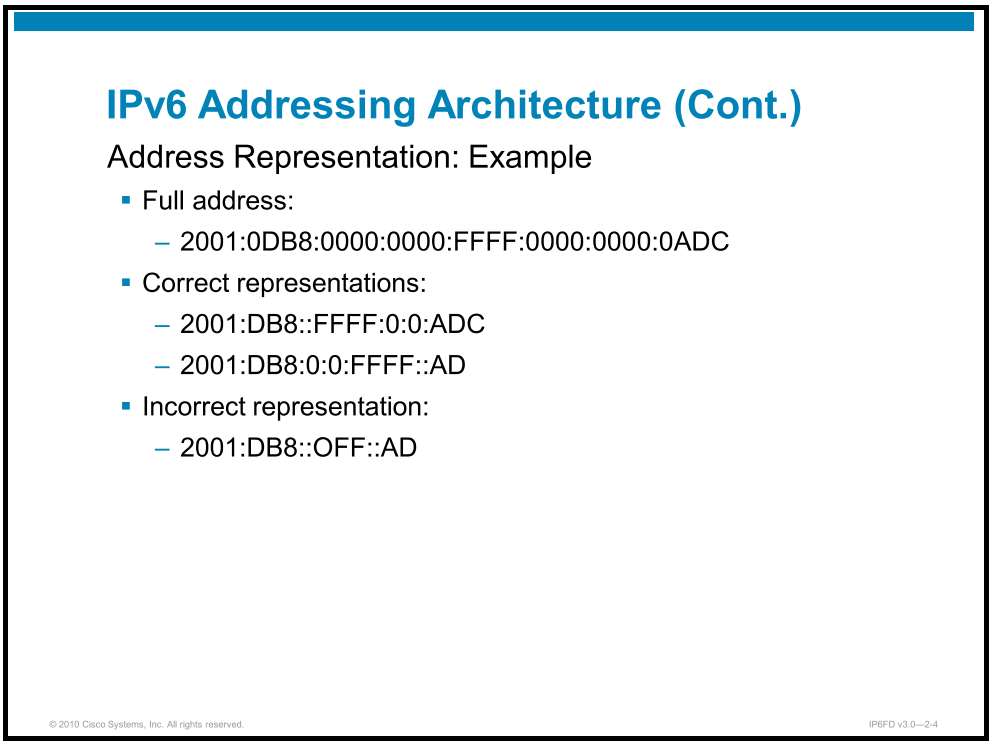


IPv6 addresses are represented as a series of eight 16-bit hexadecimal fields that are separated by colons. The A, B, C, D, E, and F in hexadecimal fields are case-insensitive.

These are some ways to shorten the writing of IPv6 addresses:

- The leading zeros in a field are optional, so 010F can be written as 10F and 0000 can be written as 0.
- Successive fields of zeros can be represented as a double colon (::), but only once in an address. An address parser can identify the number of missing zeros by separating the two parts and filling in zeros until the 128 bits are completed. However, if two double colons are placed in the address, there is no way to identify the size of each block of zeros. Therefore, only one double colon is possible in a valid IPv6 address.

The use of the double-colon technique makes many addresses very small; for example, FF01:0:0:0:0:0:0:1 becomes FF01::1. The unspecified address is written as a double colon because it contains only zeros.



This figure shows the use of the double colon to represent multiple contiguous 16-bit chunks of zeros in an IPv6 address. The last representation that is shown in the figure is incorrect: The "::" notation can appear only once in an address because multiple uses can make the address ambiguous. In that last example, the parser cannot tell whether the missing bits (four 16-bit sections) are apportioned with 16 at the first double colon and 48 at the last double colon or some other combination.

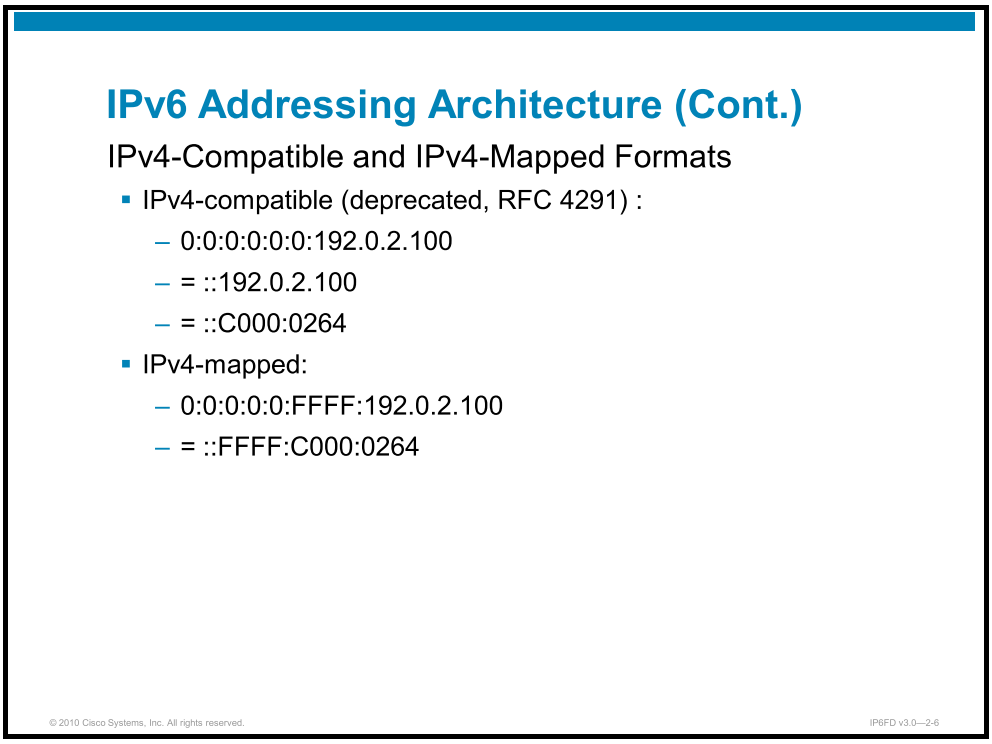
IPv6 Addressing Architecture (Cont.)

Address Representation: Further Examples

Full Address	Correct Representation
FF02:0:0:0:0:0:0:1	FF02::1
FF15:0:0:0:0:0:1:c001	FF15::1:C001
0:0:0:0:0:0:0:1	::1
0:0:0:0:0:0:0:0	::

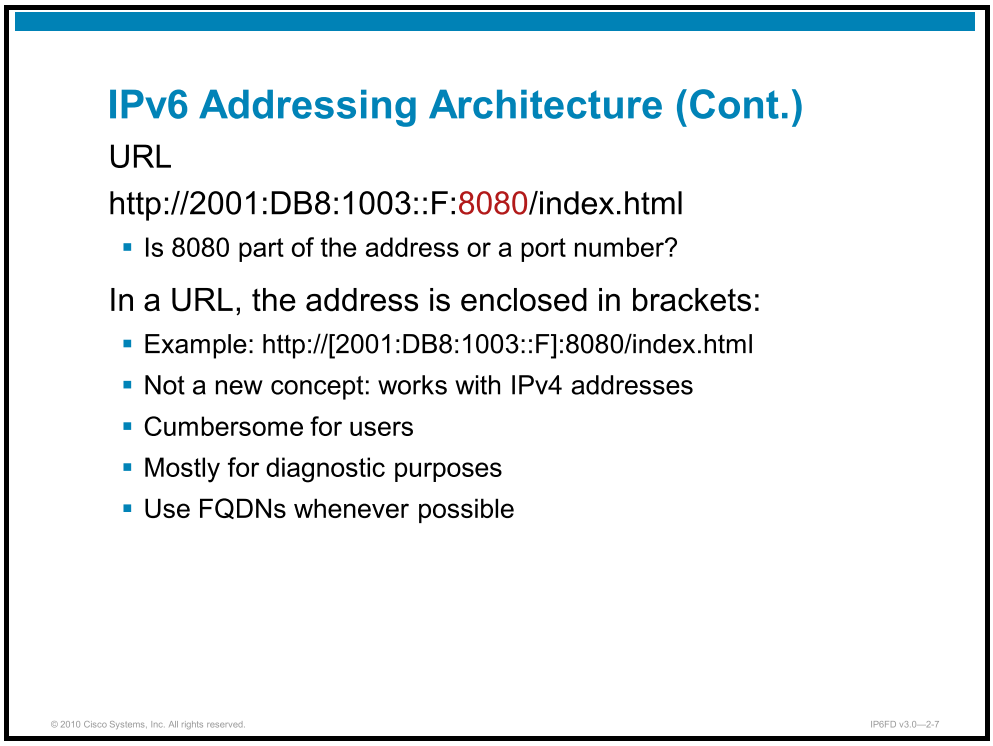
These are more examples of correct IPv6 address representations:

- FF02::1 represents all IPv6 nodes within the link-local multicast.
- FF15::1:C001 represents temporary site-local multicast.
- ::1 represents loopback.
- :: is unspecified.



To process automatic tunnels over IP version 4 (IPv4), a special IPv4-compatible IPv6 address format is defined as ::A, B, C, D, where A, B, C, and D are the IPv4 addresses in decimal form. Few systems will accept this addressing format. More often, the IPv4 address will need to be converted to hexadecimal, a transition mechanism that is not implemented in every IPv6 stack. IPv4-compatible addressing has been deprecated but is included here for reference.

The IPv4-mapped IPv6 address is used for internal system mechanisms and should never be transmitted to the network by a node. This mechanism allows systems to represent IPv4 addresses as 128-bit IPv6 addresses.

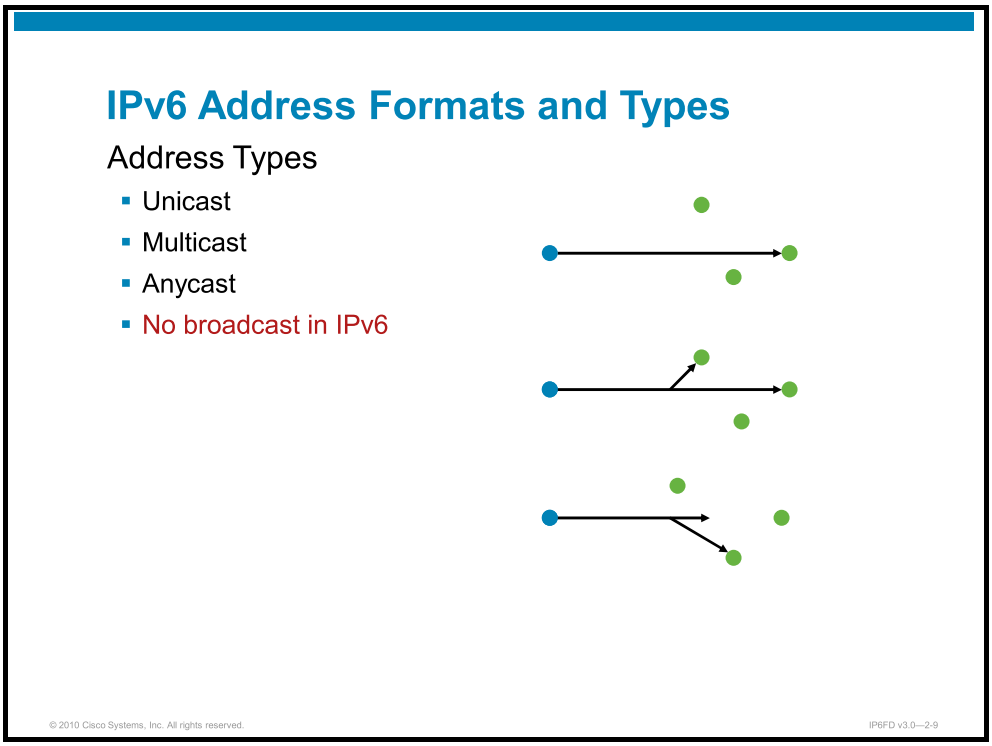


In a URL, the colon is already used to place an optional port number, as illustrated in the URL http://www.example.com:8080/index.html. A URL parser must then differentiate between the colon of a port number and the colon inside an IPv6 address, which is impossible because of the compression technique. To identify its IPv6 address while keeping the colon, the address must be enclosed between brackets, as shown in the figure.

Using IPv6 addresses inside a URL is cumbersome for users and should be used only for diagnostic purposes or when no naming service is available. Otherwise, fully qualified domain names (FQDNs) should be used in place of these literal IPv6 addresses (called literals).

IPv6 Address Formats and Types

This topic describes the format and uses of the various types of IPv6 addresses.

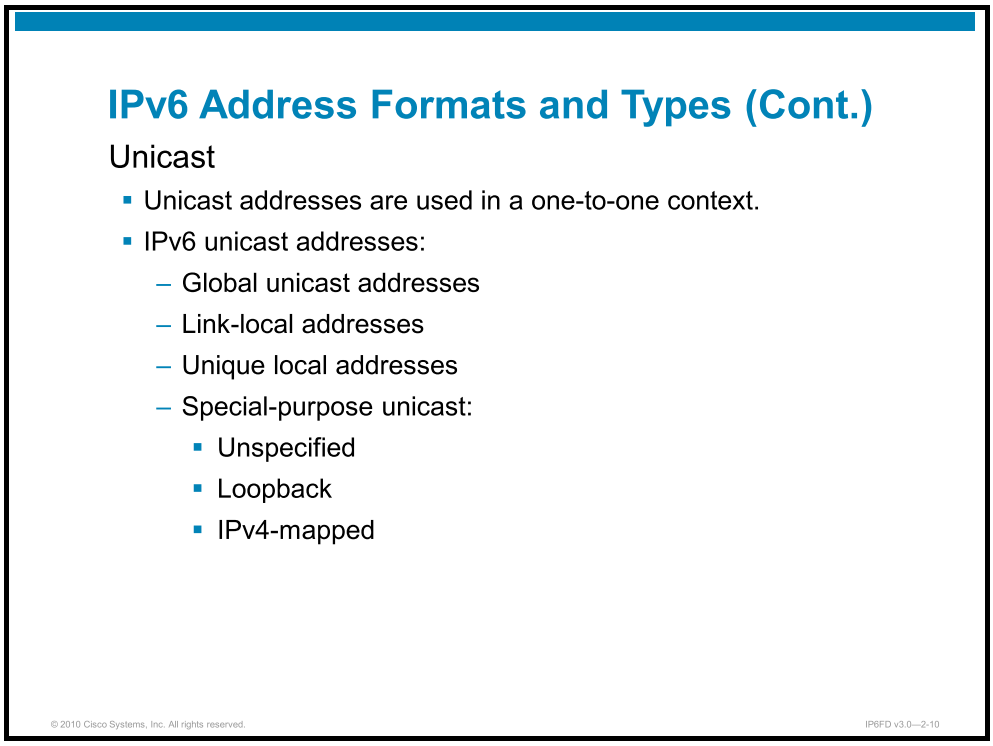


IPv6 supports three types of addresses:

- Unicast
- Multicast
- Anycast

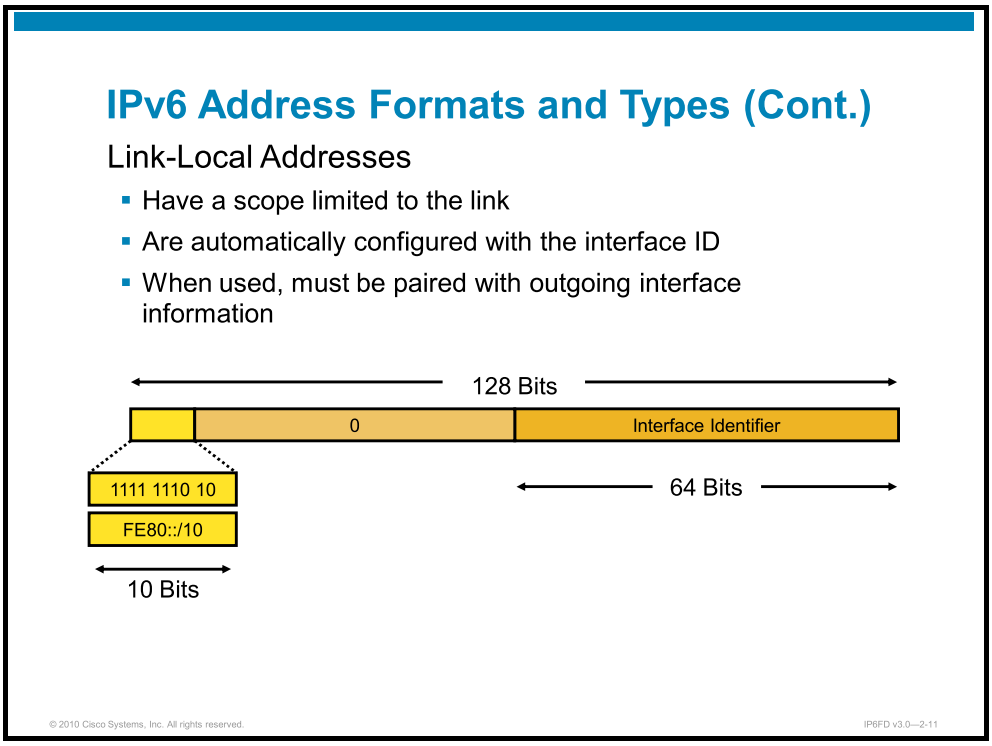
Each address type has specific rules regarding its construction and use.

IPv6 has no support for broadcast addresses in the way that they are used in IPv4. Instead, specific multicast addresses (such as the all nodes multicast address) are used.



IPv6 unicast addresses can be aggregated with prefixes of arbitrary bit length, like IPv4 addresses under classless interdomain routing (CIDR).

There are several types of unicast addresses in IPv6, including global addresses, site-local addresses (deprecated), unique local addresses, and link-local addresses. There are also some special-purpose subtypes of global unicast, such as the unspecified address, loopback address, and IPv6 addresses with embedded IPv4 addresses. Additional address types or subtypes might be defined in the future.



All IPv6-enabled interfaces must have a link-local address.

Link-local addresses are used for addressing on a single link, so they have a scope that is limited to the link. Link-local addresses are created dynamically on all IPv6 interfaces by using a specific link-local prefix, FE80::/10, and a 64-bit interface identifier.

Link-local addresses are used for automatic address configuration, neighbor discovery, and router discovery. Many routing protocols also use the addresses.

Link-local addresses can serve as a way to connect devices on the same local network, without requiring global or unique local addresses.

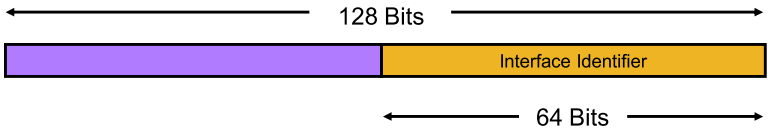
When communicating with a link-local address, you must specify the outgoing interface because every interface connects to FE80::/10.

Tip IPv6 has a 128-bit address space, but 64 bits are used for the host number on the subnet. A better way to look at the address space is to say that IPv6 supports 2^{64} subnets, and each subnet can have a practically unlimited number of hosts. In any case, there are more than enough networks and hosts for the future.

IPv6 Address Formats and Types (Cont.)

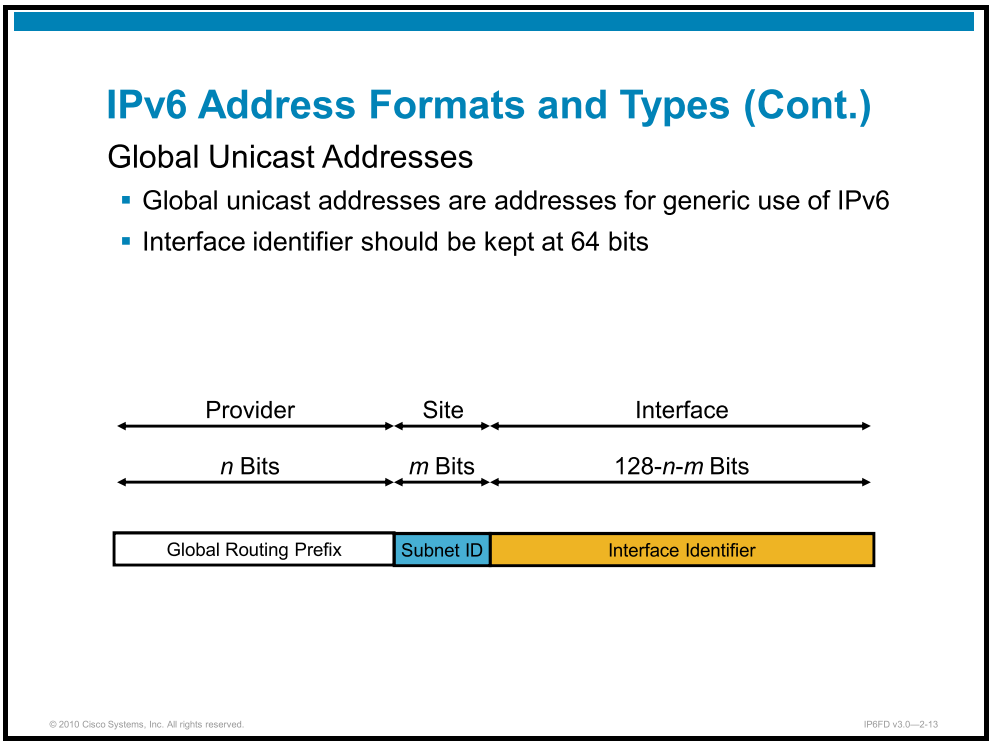
Interface Identifiers

- Used to identify interfaces on a link:
  - Must be unique on that link
  - Can be globally unique
- Unicast addresses should have a 64-bit interface ID:
  - Except for unicast addresses that start with binary 000
  - Interface ID constructed in modified EUI-64 format



Interface identifiers in IPv6 unicast addresses are used to identify interfaces on a link. The identifiers can also be thought of as the host portion of an IPv6 address and must be unique on that link. These identifiers may also be unique over a broader scope: When the identifier is derived directly from the data link layer address of the interface (for example, IEEE 802 MAC), the scope of that identifier is assumed to be universal (global).

Interface identifiers are always 64 bits and can be created dynamically, based on Layer 2 addresses such as Ethernet MAC addresses.



Global unicast addresses correspond to the principal use of IPv6 addresses for generic global IPv6 traffic and consume the most important part of the address space.

The structure of a global unicast address is as follows:

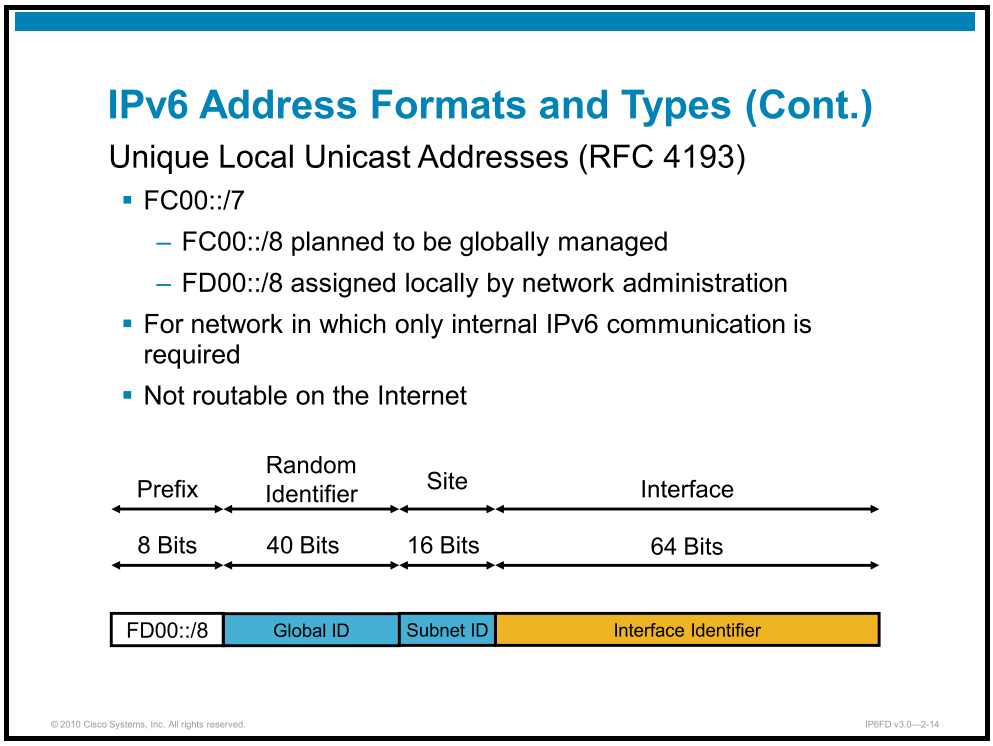
- A global routing prefix, typically a /48, is assigned to a site.
- A subnet identifier, typically 16 bits, is used to identify links within a site.
- A 64-bit interface identifier identifies the interface of the node.

The interface identifier can be of arbitrary length but should be kept at 64 bits for several reasons:

- Stateless autoconfiguration of hosts depends on the 64-bit length of the interface identifier.
- Some operating systems, such as Microsoft Windows XP, do not allow the changing of the default network mask.
- Because of the greater length of addresses in IPv6, some hardware platforms might be limited to hardware-assisted forwarding of prefixes that are as long as 64 bits. Longer prefixes are processed in software.

Examples of global addresses can be found in RFC 3587, *IPv6 Global Unicast Address Format.* The structure that is proposed in this document provides for aggregation of routing prefixes to limit the number of entries in the global routing table.

Later in the course, you will see how the Internet Assigned Numbers Authority (IANA) and the Regional Internet Registries (RIRs) allocate IPv6 global unicast address space from the range of addresses that start with binary value 001 (2000::/3).



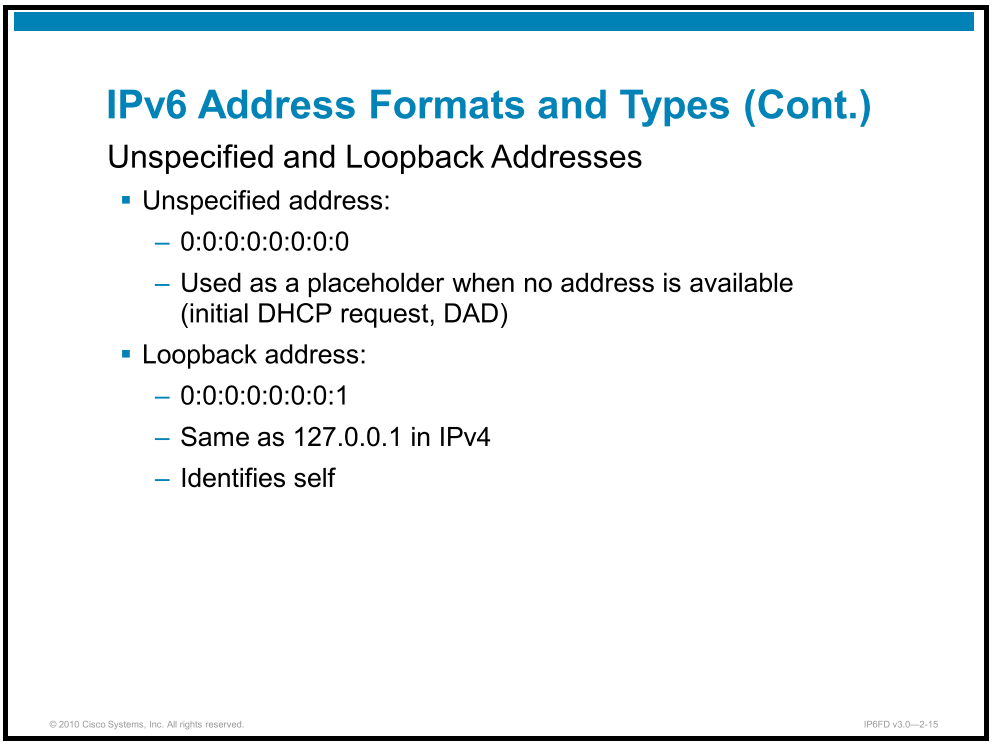
Unique local addresses were designed as a replacement for site-local addresses, specifically to resolve some scoping issues. Unique local addresses have a site-specific scope but are almost globally unique; that is, they are highly unlikely to have an address-space clash.

The structure of a unique local address is as follows:

- The FD00::/8 prefix:
  - Indicates a locally assigned unique local address
  - Is reserved for possible use as a centrally registered unique local address
- A 40-bit, pseudo-random global ID; the least significant 40 bits from the result of Secure Hash Algorithm 1 (SHA-1) (64-bit time of day + extended unique identifier [EUI]-64)
- A 16-bit subnet ID to identify the subnet within the site
- A 64-bit interface identifier

Unique local addresses are defined in RFC 4193, *Unique Local IPv6 Unicast Addresses*. These addresses are used specifically to address implementation problems with the use of site-local addresses, as well as address space clashes that such use might cause. Unique local addresses also provide an IP addressing mechanism for organizations that prefer the concept of private address space for most internal communications and as part of their security policy architecture.

Tip Today, many companies use RFC 1918 addresses within their organizations. Network engineers tend to gravitate toward the 10.0.0.0/8 reserved block. This practice leads to problems when companies merge. The Internet Engineering Task Force (IETF) was concerned that the same issue would arise with site-local addresses, so it designed unique local addresses to introduce a large random component into the nonroutable prefix space. There is almost no chance of a prefix collision when two merging companies use properly self-allocated unique local address prefixes.



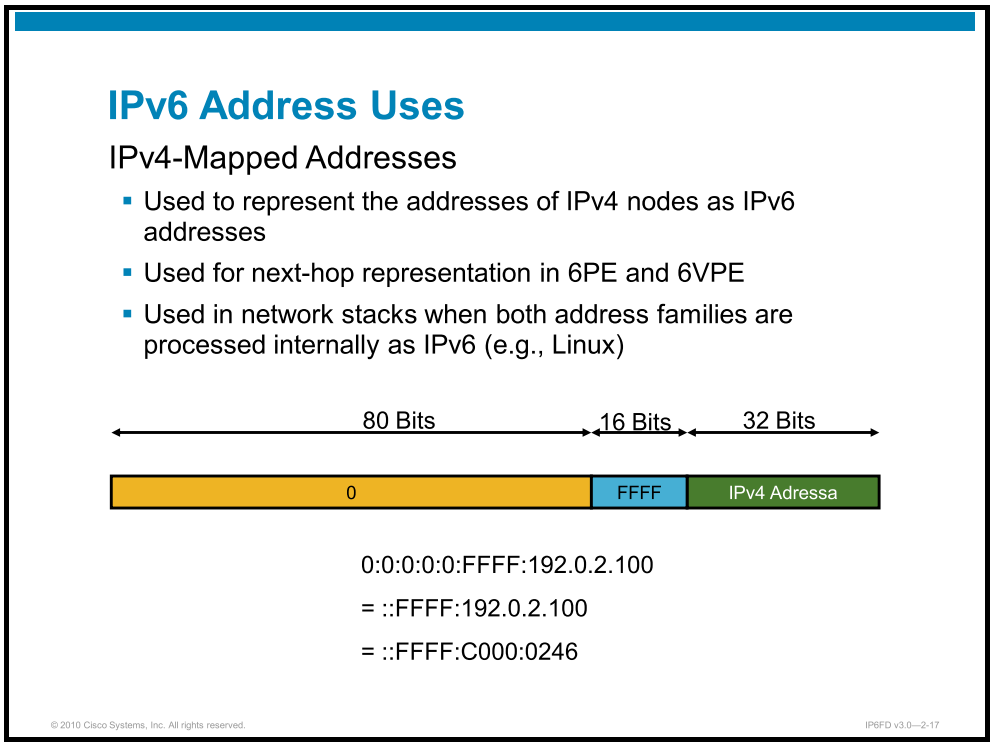
An unspecified address is 0:0:0:0:0:0:0:0, or simply “::”.

An unspecified address is used on a network only as a source address for special purposes. An unspecified address is a placeholder when no address is available. For example, an unspecified address is used when a host requests an address to a DHCP server or when a Duplicate Address Detection (DAD) packet is sent. An unspecified address should never be the source address of an IPv6 packet, and routers must not forward packets with an unspecified source.

The loopback address identifies a local interface in the IP stack. This address is the IPv6 equivalent of the IPv4 127.0.0.1 loopback. The address is 0:0:0:0:0:0:0:1, or simply ::1.

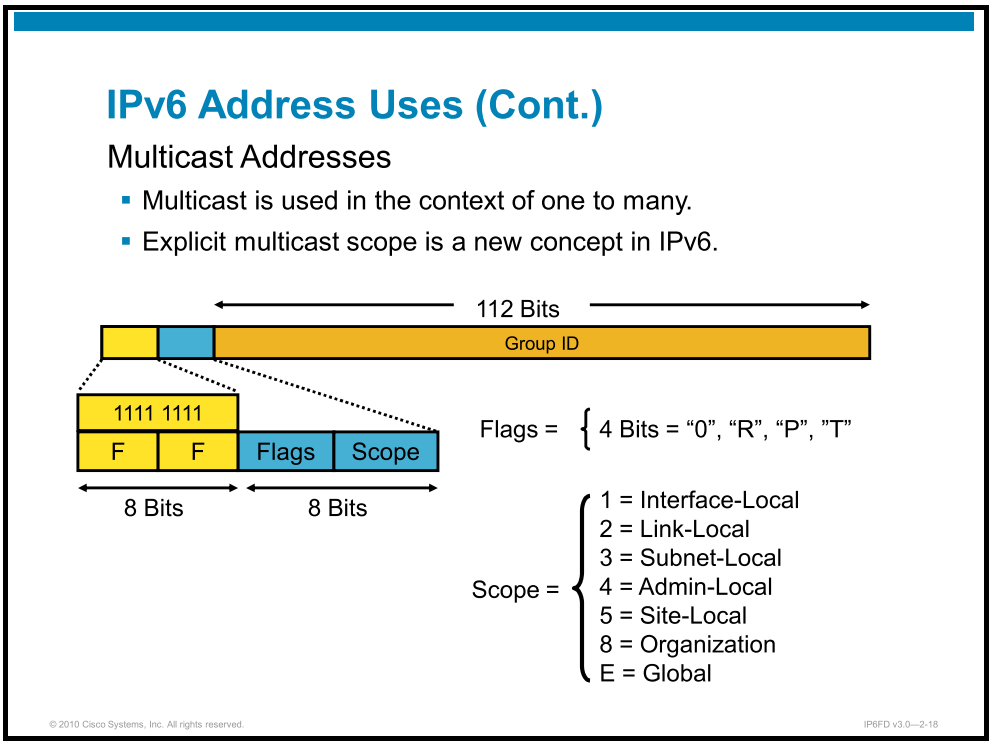
IPv6 Address Uses

This topic describes how to create and use the various types of IPv6 addresses.



IPv4-mapped addresses are IPv6 addresses that represent an IPv4 address. On a dual-stack node (a node that supports both IPv6 and IPv4), an IPv6 application that sends traffic to a destination that is represented by an IPv4-mapped IPv6 address will send IPv4 packets to that IPv4 destination.

In most cases, the IPv4-mapped addresses are used inside the dual-stack node application programming interface (API; see RFC 2133). RFC 2765, *Stateless IP/ICMP Translation Algorithm (SIIT)*, specifies a transition mechanism in which IPv4-mapped addresses are used in IPv6 packets. Dual-stack nodes often internally treat IPv4 addresses as IPv4-mapped addresses, to process IP addresses as 128 bits. Syslog entries on a dual-stack system that logs IPv4 addresses are often logged as IPv4-mapped 128-bit addresses. These entries are also used for next-hop representation with Cisco IPv6 Provider Edge (6PE) and IPv6 Virtual Private Network (VPN) Provider Edge (6VPE) routers, when an IPv4 address is used for the next hop of an IPv6 prefix.



A multicast address identifies a group of interfaces. Traffic that is sent to a multicast address is sent to multiple destinations at the same time. An interface may belong to any number of multicast groups. Multicast is used in the core of many functions in IPv6.

Multicast addresses are defined by the prefix FF00::/8. The second octet defines the flags and the scope of the multicast address. Flags are defined as 0RPT, and these conditions apply:

- 0 is reserved and must equal 0.
- R indicates rendezvous point and is almost always set to 0.
- P indicates prefix dependency and is almost always set to 0.
- T is the temporary bit. For a temporary multicast address, T equals 1; for a permanent multicast address, T equals 0.

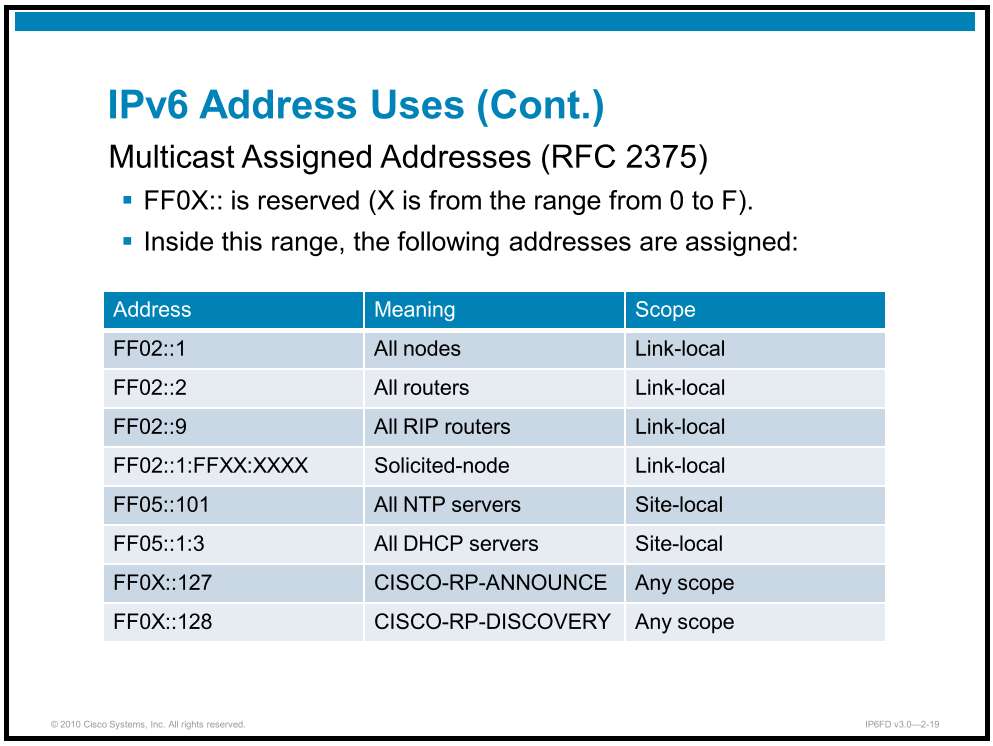
Note If R equals 1, P and T must also equal 1.

The scope parameter equals 1 for the scope of the interface (loopback transmission); 2 for the link scope (like the unicast link-local scope); 3 for the subnet-local scope, in which subnets may span multiple links; 4 for the administrative-local scope (administratively configured); 5 for the site scope; 8 for the organizational scope (multiple sites); and E for the global scope. For example, a multicast address that starts with FF02::/16 is a permanent multicast address with a link-local scope.

The lower 112 bits of the multicast address constitute the multicast group ID.

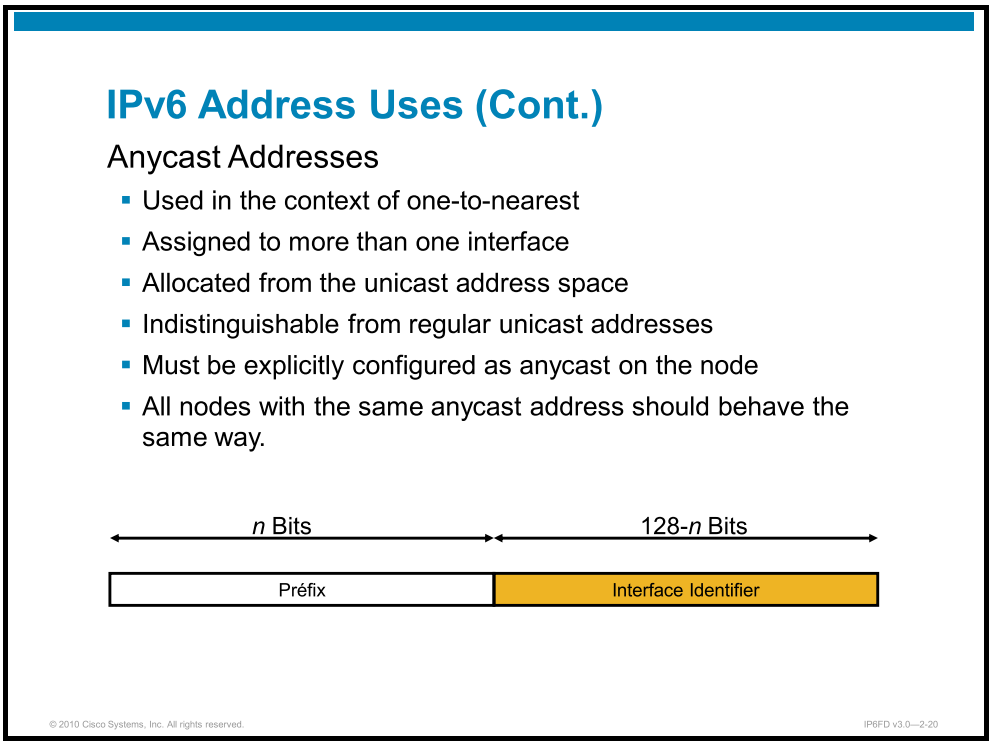
Multicast is frequently used in IPv6 and replaces broadcast. There is no broadcast in IPv6.

There is no Time to Live (TTL) in IPv6 multicast. The scoping is defined inside the address.



The multicast addresses FF00:: to FF0F:: are reserved. Inside that range, RFC 2375 assigns these addresses, among others:

- FF02::1 represents all nodes on the link-local scope.
- FF02::2 represents all routers on the link-local scope.
- FF02::9 represents all Routing Information Protocol (RIP) routers on the link.
- FF02::1 represents FFXX:XXXX: solicited-node.
- FF05::101 represents all Network Time Protocol (NTP) servers on the site-local scope.
- FF05::1:3 represents all DHCP servers in site.
- FF0X::127 represents CISCO-RP-ANNOUNCE (multicast rendezvous point).
- FF0X::128 represents CISCO-RP-DISCOVERY.



An IPv6 anycast address is assigned to an interface on more than one node. When a packet is sent to an anycast address, it is routed to the nearest interface that has that address. The nearest interface is found according to the measure of distance of the particular routing protocol. All nodes that share the same address should behave the same way so that the service is offered similarly regardless of the node that services the request.

The idea of anycast in IP was proposed in 1993. For IPv6, anycast is defined as a way to send a packet to the nearest interface that is a member of the anycast group. This technique enables a type of mechanism that can discover the nearest node of a specific group.

Anycast addresses are allocated from the unicast address space, so they are indistinguishable from the unicast address. When the anycast addresses are assigned to a node interface, the node must be explicitly configured to know that the address is an anycast address.

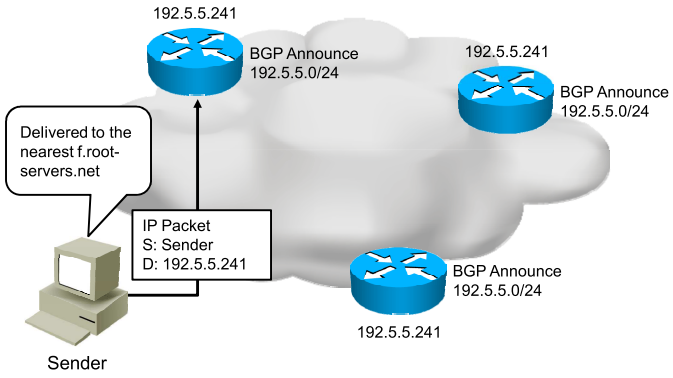
There is little widespread experience with anycast usage. The router-subnet anycast and the Mobile IPv6 home agent anycast are among the few anycast addresses that are currently assigned.

Tip The root Domain Name System (DNS) servers use IPv4 anycast. There are 13 root server addresses, but a much larger number of widely dispersed hosts provide DNS services. Anycast is a powerful function of IP networks and is probably underused today.

IPv6 Address Uses (Cont.)

IPv4 Anycast Example

- BGP determines which destination will receive traffic.
- Used for root DNS servers, 6to4 relays, etc.



This example shows how anycast is being used on the IPv4 Internet.

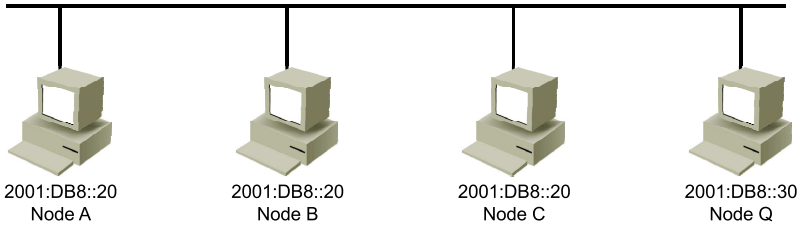
The IPv4 address of the F root name server is 192.5.5.241. The IPv4 network 192.5.5.0/24 is advertised by a few autonomous systems across the Internet. A packet that is destined for 192.5.5.241 will go to the nearest advertising autonomous system, based on the Border Gateway Protocol (BGP) routing table. Therefore, more than one physical server and network can provide the same service in different geographical locations, making the service more resilient to faults.

This example also applies for IPv6. The IPv6 address of the same server is 2001:500:2F::F.

IPv6 Address Uses (Cont.)

Anycast Addresses: LAN

- Nearest anycast address is whichever host is put into Node Q neighbor cache first.
- DAD is not done for these addresses.



This example shows how, in IPv6, anycast addressing can be accomplished on-link. When an address is assigned to an interface, an option or switch specifies that the address is an anycast address. When an anycast address is applied to an interface, that interface processes neighbor discovery differently (such as suppressing DAD) to accommodate duplicate addresses on-link.

Required IPv6 Addresses

This topic describes how to determine the required IPv6 addresses for an IPv6 host and an IPv6 router.

Required IPv6 Addresses

Required Host Addresses

An IPv6 host interface requires the following IPv6 addresses for proper operation:

- A link-local address
- Loopback address (::1)
- All-nodes multicast address (FF02::1)
- Any additional unicast and anycast addresses configured (automatically or manually)
- Solicited-node multicast address for each of its unicast and anycast addresses
- Multicast addresses of all other groups to which the host belongs

An IPv6 host has many IPv6 addresses, all of which are used in different contexts. This is a good example of the power of IPv6. Its address space enables efficient use of addresses for protocol design.

Solicited-node multicast addresses are discussed with the neighbor discovery protocol, later in the course.

Required IPv6 Addresses (Cont.)

Required Router Addresses

An IPv6 router interface requires the following IPv6 addresses for proper operation:

- All of the required host addresses
- All router multicast addresses (FF02::2)
- Other unicast or anycast-configured addresses

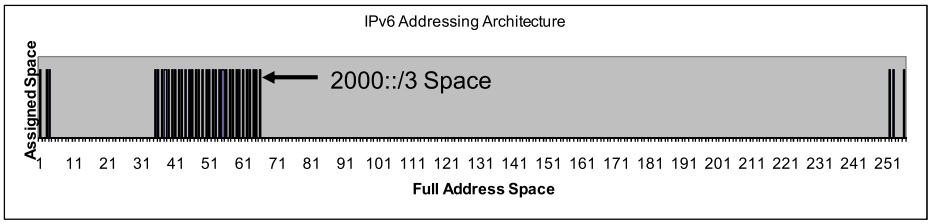
An IPv6 router is first an IPv6 node, so it has all the required host addresses. In its function as a router, it has additional addresses, as listed in the figure.

Required IPv6 Addresses (Cont.)

Addressing Architecture

The graph shows the IANA assignments of IPv6 addresses:

- X axis = Full address space
- Y axis = 0 or 1 if space is allocated
- 1 does not mean that space is used, only that it is reserved by IANA



In the figure, the X axis represents the complete address space, divided into 256 parts, from 1 to 256. The Y axis is binary and is 1 when the space is assigned.

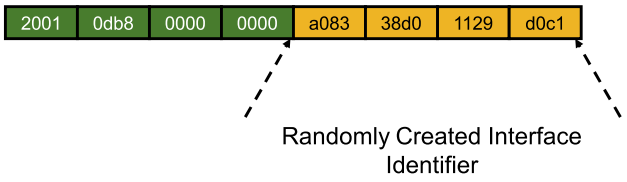
The assigned address range is not related to the actual allocation to users and networks:

- 0::/3 represents the special unicast address space (unspecified, loopback, IPv4-compatible, and IPv4-mapped addresses).
- 2000::/3 is the range from which the IANA allocates IPv6 unicast addresses. That range represents 1/8 of the address space and is shown as the largest black part on the chart.
- FC00::/7 is the unique local unicast address range, which uses 1/128 of the address space and is shown by the narrow black line near the right side of the chart.
- FE80::/10 is the link-local unicast address range, which uses 1/1024 of the address space. This range is too small to be seen on the chart but would be at the rightmost end.
- FEC0::/10 is the site-local unicast address range, which uses 1/1024 of the address space. This range is too small to be seen on the chart but would be at the rightmost end. Although site-local addresses have been deprecated, this address space is reserved for the foreseeable future, to prevent compatibility problems.
- FF00::/8 is the multicast address range, which uses 1/256 of the address space and is shown as the last small black line at the rightmost end of the chart.

Required IPv6 Addresses (Cont.)

Privacy Interface Identifier

- Defined in RFC 3041
- Supports randomly generated interface identifier
- Hides hardware information from network layer
- Can be permanent or temporary
- Temporary random interface ID prevents global device tracking.

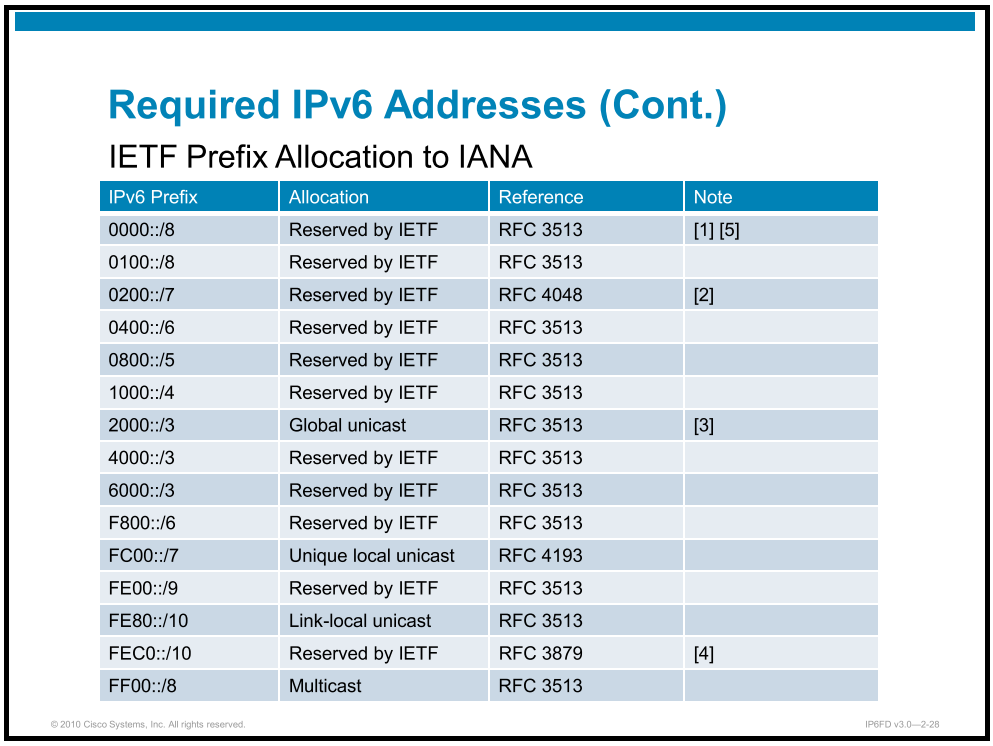


A process for generating a random interface identifier is provided in RFC 3041. Windows XP implements this process by default and prefers to use this address for outgoing communication.

The randomly generated interface identifier address has a short lifetime and is regenerated periodically (once per day in Windows).

This process is considered a privacy extension because, without it, if you create an interface identifier from a MAC address, others could track your activity and point of connection. To address privacy concerns that are associated with this level of tracking, the privacy extension was created.

Tip Consider the situation for a business traveler. As travelers move around the world, connecting to various Internet hosts from different IPv6 subnets, they leave a telltale signature—their 64-bit interface identifier. This identifier is consistent over time, even as the subnet prefix changes, as long as the node is using autoconfiguration, which incorporates the burned-in MAC address of the underlying interface. Privacy addresses were invented to allow the interface identifier, which must simply be unique on the subnet, to also change over time and in unpredictable ways.



The notes that are referenced in the figure are defined here:

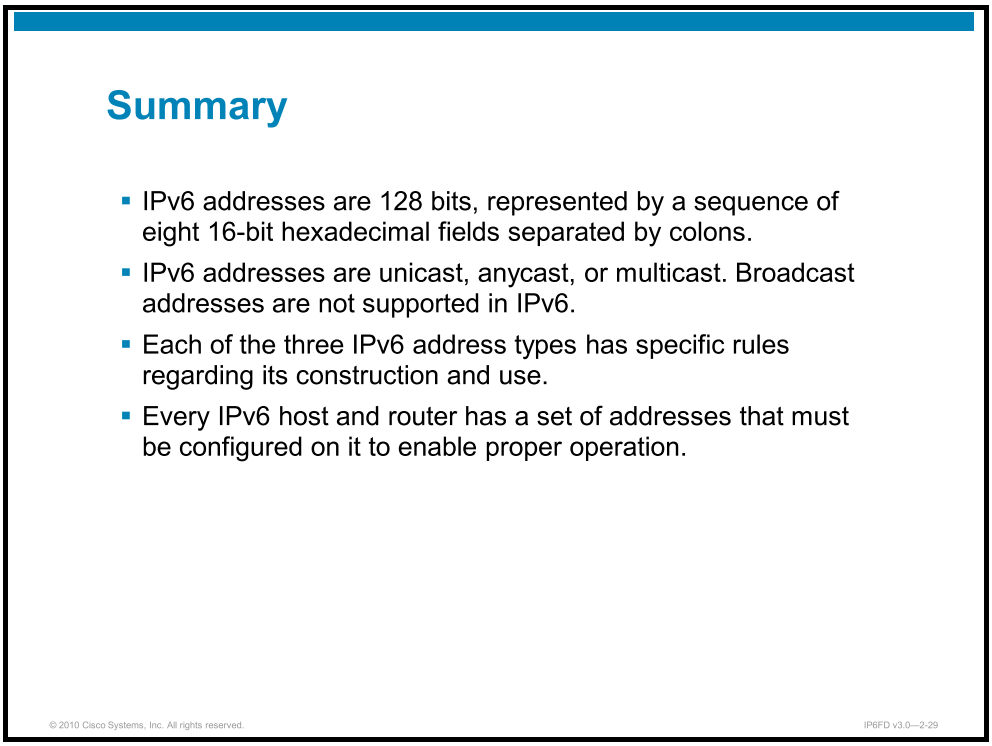
- [1] The unspecified address, the loopback address, and the IPv6 addresses with embedded IPv4 addresses are assigned out of the 0000::/8 address block.
- [2] 0200::/7 was previously defined as an Open Systems Interconnection (OSI) network service access point (NSAP)-mapped prefix set (RFC-gray-rfc1888bis-03.txt). This definition was deprecated in December 2004 (RFC 4048).
- [3] The IPv6 unicast space encompasses the entire IPv6 address range except for FF00::/8 (RFC 3513). IANA unicast address assignments are currently limited to the IPv6 unicast address range of 2000::/3. IANA assignments from this block are registered in the IANA registry: iana-ipv6-unicast-address-assignments.
- [4] FEC0::/10 was previously defined as a site-local scoped address prefix. This definition has been deprecated as of September 2004 (RFC 3879).
- [5] 0000::/96 was previously defined as the IPv4-compatible IPv6 address prefix. This definition was deprecated by RFC 4291.

The IPv6 assignments, from IETF to IANA, are shown in the figure. IANA then passes prefixes to the regional registries on an as-needed basis for unicast space. For multicast, IANA may make individual assignments as requested.

Tip IETF makes assignments to IANA in RFCs. Presumably, much of the unallocated space will be assigned to IANA as additional unicast prefixes are required, but by holding most space in reserve until it is needed, IETF does not preclude innovation for the protocol. Perhaps a new type of prefix or a radically different use of the addressing space will be developed.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 2

Describing the IPv6 Header Format

Overview

The header format for each IP packet carries crucial information for the routing and processing of each packet payload. Header construction also plays an important role in the efficiency and extensibility of the network. This lesson describes the IP version 6 (IPv6) header format and identifies the major changes in header construction from IP version 4 (IPv4) to IPv6, which will yield long-term benefits to the Internet.

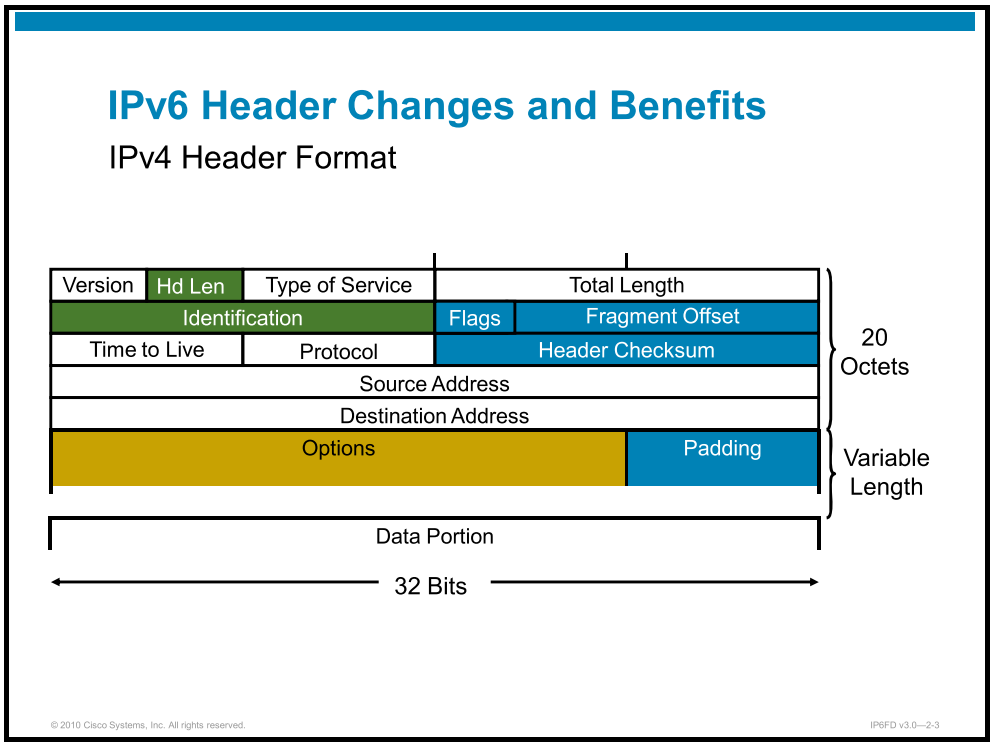
Objectives

Upon completing this lesson, you will be able to describe changes in the IPv6 header and the purpose of extension headers. This ability includes being able to meet these objectives:

- Describe the major changes in an IPv6 header and the benefits of these changes
- Describe the new fields that were added to an IPv6 header and explain their operation
- Describe the purpose of extension headers in IPv6

IPv6 Header Changes and Benefits

This topic describes the major changes in an IPv6 header and the benefits of these changes.



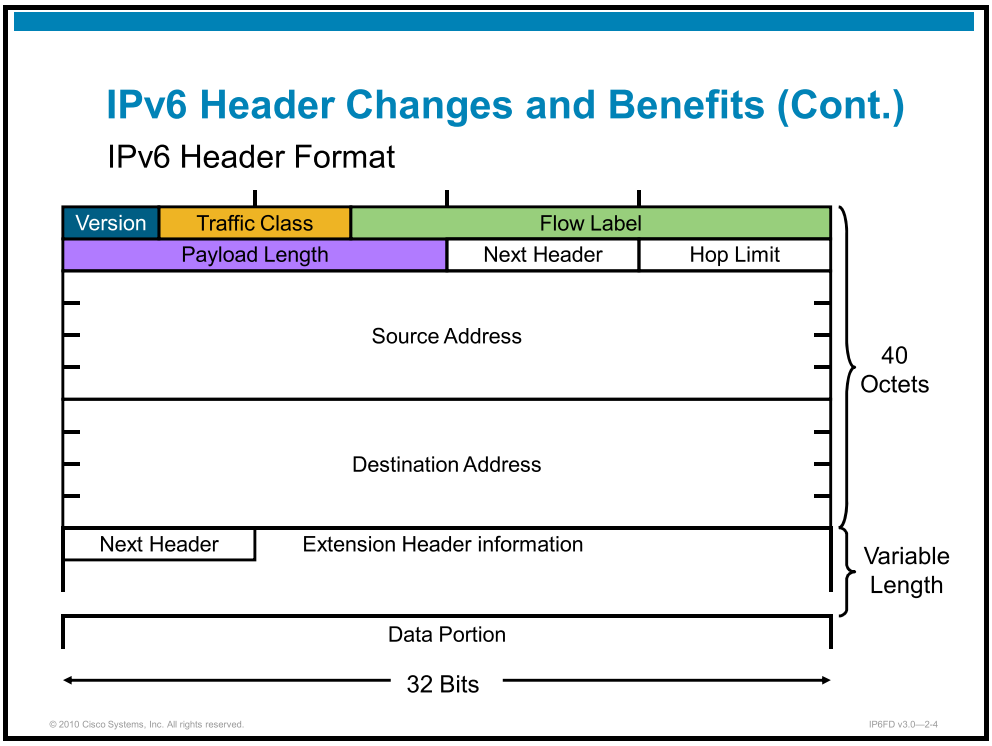
The IPv4 header contains 12 fields. Following those fields is an Options field of variable length, which the figure shows in yellow, and a data portion, which is usually the transport layer segment. The basic IPv4 header has a size of 20 octets. The Options field increases the size of the IP header.

Of these 12 header fields, 6 are removed in IPv6; these fields are shown in green and blue in the figure. The main reasons for removing these fields in IPv6 are as follows:

- The Internet Header Length (Hd Len) field was removed because all IPv6 headers are a fixed, 40-byte length, unlike IPv4, in which the header length is variable.
- Fragmentation is now processed differently and does not need the fields in the basic IP header. In IPv6, routers no longer process fragmentation, a change that removes the processing issues that result when routers process IPv4 fragmentation. The related, removed fields appear in the Fragmentation Extension Header in IPv6, which is attached only to a packet that is actually fragmented.
- The Header Checksum field at the IP layer was removed because most data link layer technologies already perform checksum and error control and because the relative reliability of the data link layer is very good. However, this removal forces the upper-layer optional checksums, such as User Datagram Protocol (UDP), to become mandatory.

The Options field is changed in IPv6 and is now processed by an extension header chain.

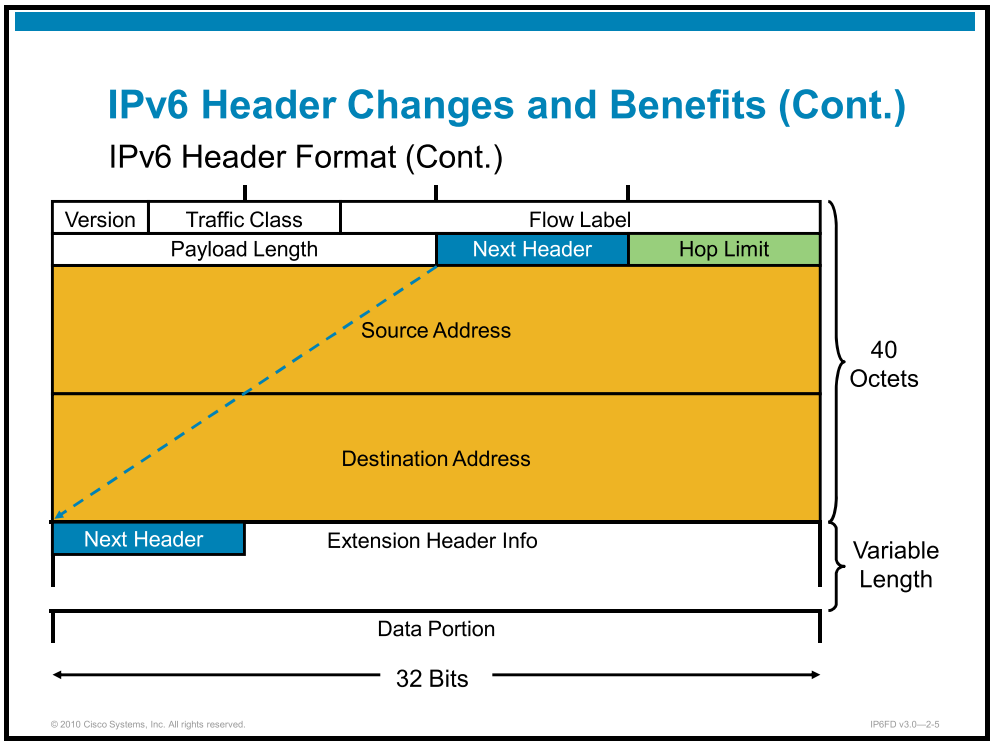
Most other fields were either unchanged or changed only slightly.



The IPv6 header has 40 octets, instead of 20 octets as in IPv4. The IPv6 header has fewer fields, and the header is aligned on 64-bit boundaries to enable fast processing by current and next-generation processors. Address fields are four times larger than in IPv4.

The IPv6 header contains eight fields:

1. **Version:** This 4-bit field contains the number 6, instead of the number 4 as in IPv4.
2. **Traffic Class:** This 8-bit field is like the type of service (ToS) field in IPv4. IPv6 nodes can mark the packet with a traffic class that can be used in differentiated services. Differentiated services functionalities are the same in IPv6 and IPv4.
3. **Flow Label:** This new field has a length of 20 bits and is used to mark individual traffic flows with unique values, which routers can use to provide per-flow nondefault treatment.
4. **Payload Length:** This field is like the Total Length field of IPv4, but because the IPv6 base header is a fixed size, this field describes the length of the payload only, not of the entire packet.



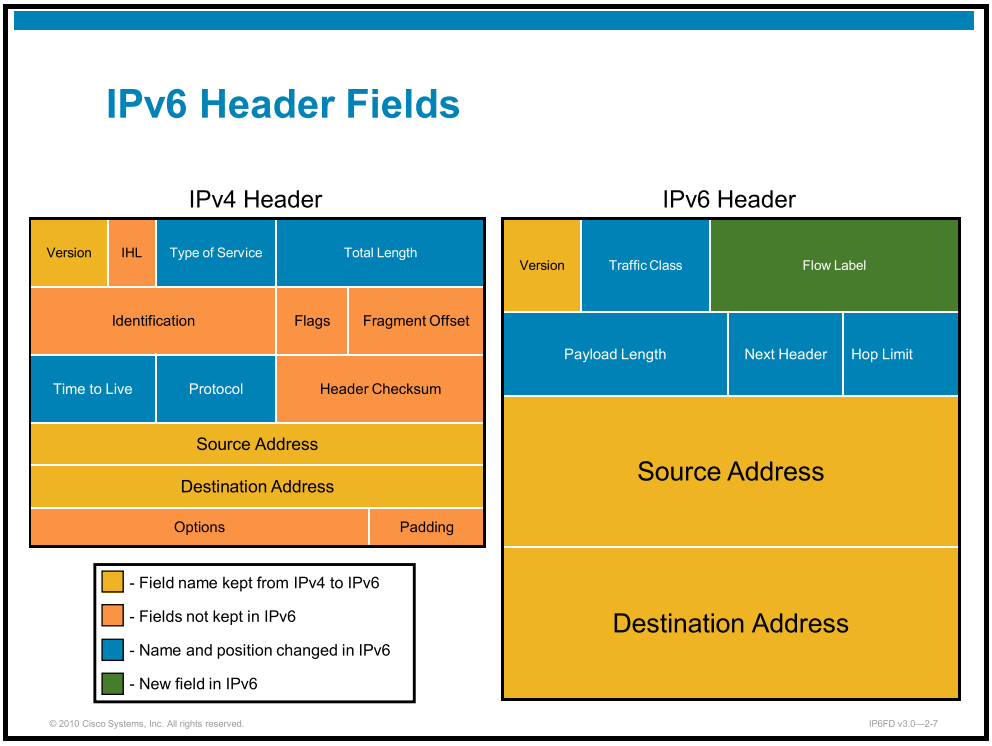
5. **Next Header:** The value of this field determines the type of information that follows the basic IPv6 header. This field can be a transport layer packet, such as TCP or UDP, or it can be an extension header, as shown in the figure. The Next Header field is like the Protocol field of IPv4 but has been renamed to reflect the more general usage: It may point to a Layer 3 IPv6 extension header rather than a Layer 4 protocol.
6. **Hop Limit:** This field specifies the maximum number of hops that an IP packet can traverse. Each hop or router will decrease this field by one. Because there is no checksum in the IPv6 header, packets can be routed more quickly through the core of the network.
7. **Source Address:** This field of 16 octets or 128 bits identifies the source of the packet.
8. **Destination Address:** This field of 16 octets or 128 bits identifies the destination of the packet.

Tip	IPv6 renames the Time to Live field to Hop Limit because the term more closely reflects the way that the field is used. The original design of IP (as described in RFC 791) called for all clocks on the network to be synchronized and for the Time to Live (TTL) to be measured in seconds. This design was later seen as impractical, and the TTL field began to be used as a simple hop counter, in which packets are discarded if the hop count is decremented to zero before final delivery.

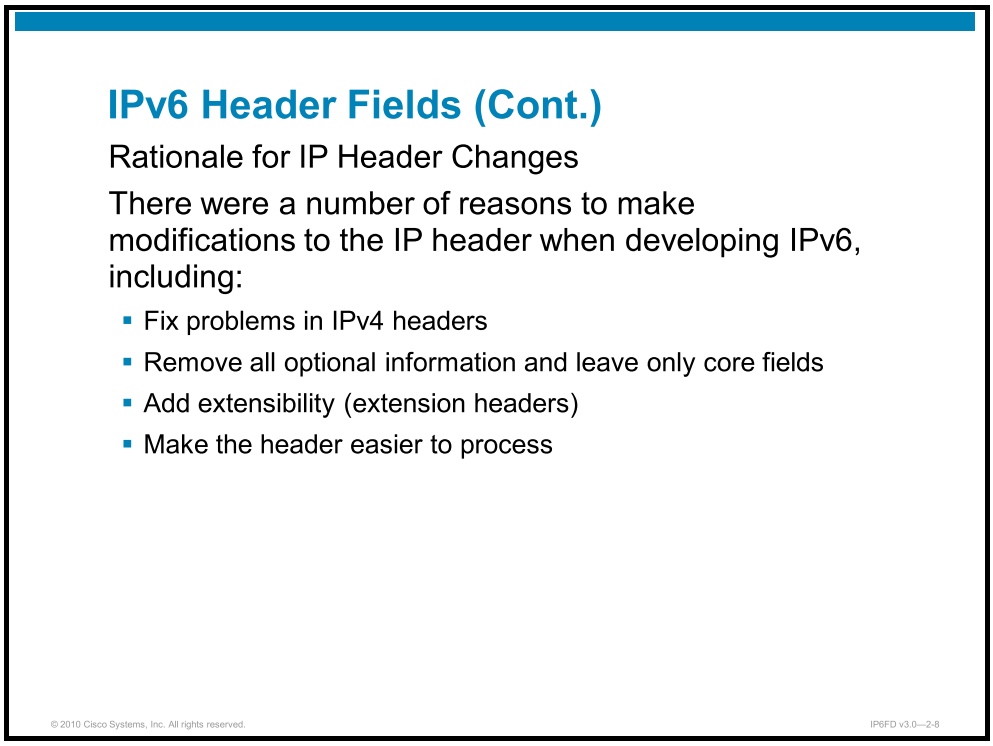
Following these eight fields are the extension headers, if any. The number of extension headers is not fixed, so the total length of the extension header chain is variable.

IPv6 Header Fields

This topic describes new fields that have been added to an IPv6 header and explains their operation.



As shown in the figure, the number of fields in the IPv6 header has decreased significantly from the number of fields in the IPv4 header.

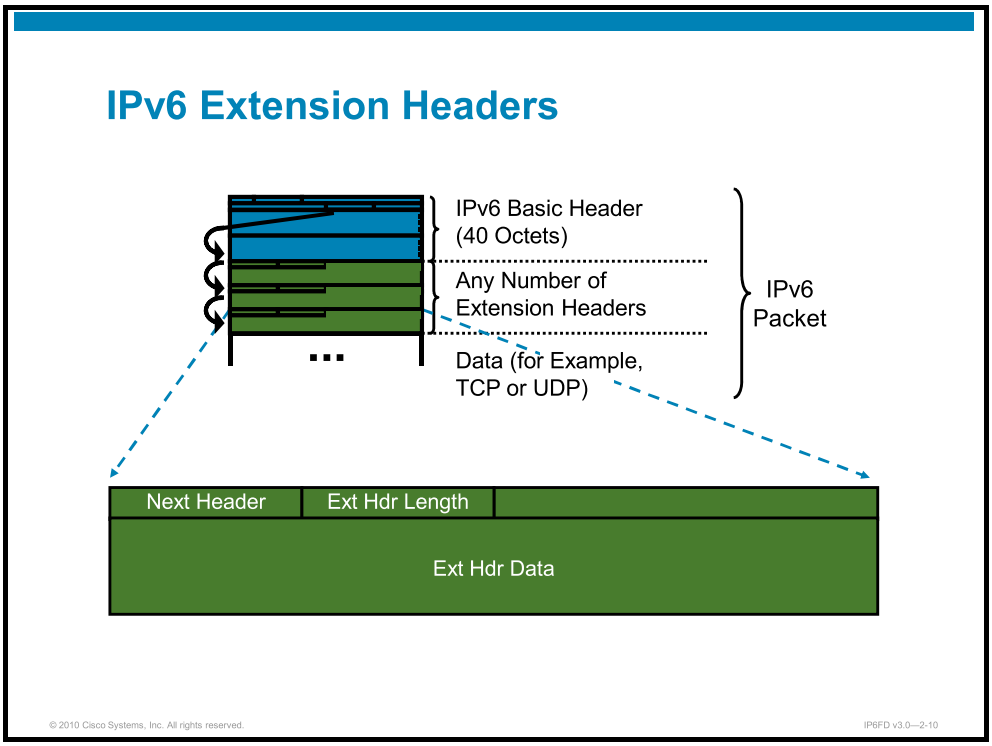


By the time that a need to develop a successor to IPv4 arrived, more than 10 years of Internet usage and growth had occurred. The designers of IPv6 took the opportunity to correct many of the problems that had arisen during the previous decade.

Problems to be corrected included removing fields that were rarely used (such as fragmentation) from the base header, adding a way to easily extend the protocol (extension headers), and streamlining the process that routers must go through to route and forward packets (eliminating the checksum). In addition, the new protocol needed to be compatible with IPv4. Using the same version field minimizes the changes at Layer 2, which is needed to carry IPv6 traffic.

IPv6 Extension Headers

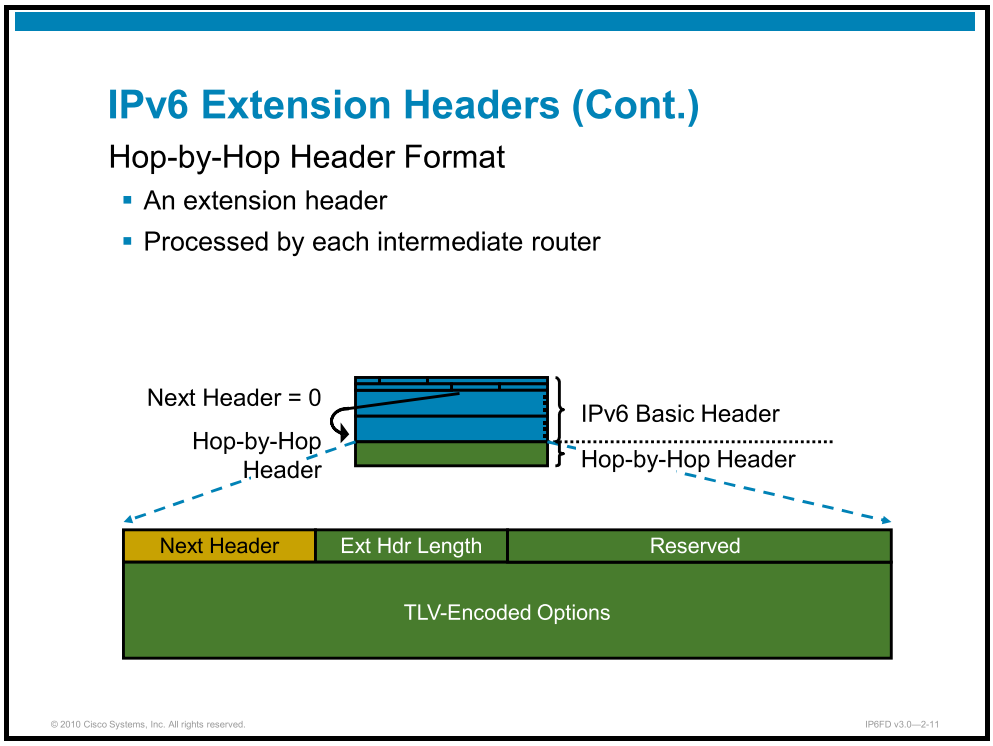
This topic describes the purpose of extension headers in IPv6.



The extension headers are optional headers that follow the IPv6 basic header. Each extension header is 8 octets (64 bits), aligned. Together, all the extension headers form a chained list of headers. Each extension header is identified by the Next Header field of the previous header. For typical applications, the final extension header will have a Next Header field that points to a transport layer protocol such as TCP or UDP. When multiple extension headers are used in the same packet, the order of the headers should be as follows:

1. Hop-by-Hop Options header
2. Destination Options header (when the Routing header is used)
3. Routing header
4. Fragment header
5. Authentication header (AH)
6. Encapsulating Security Payload (ESP) header
7. Mobility header
8. Destination Options header (when the Routing header is not used)
9. Upper-Layer header

Note The source node should follow this order, but destination nodes must be prepared to receive in any order—except for Hop-by-Hop header, which must be first.

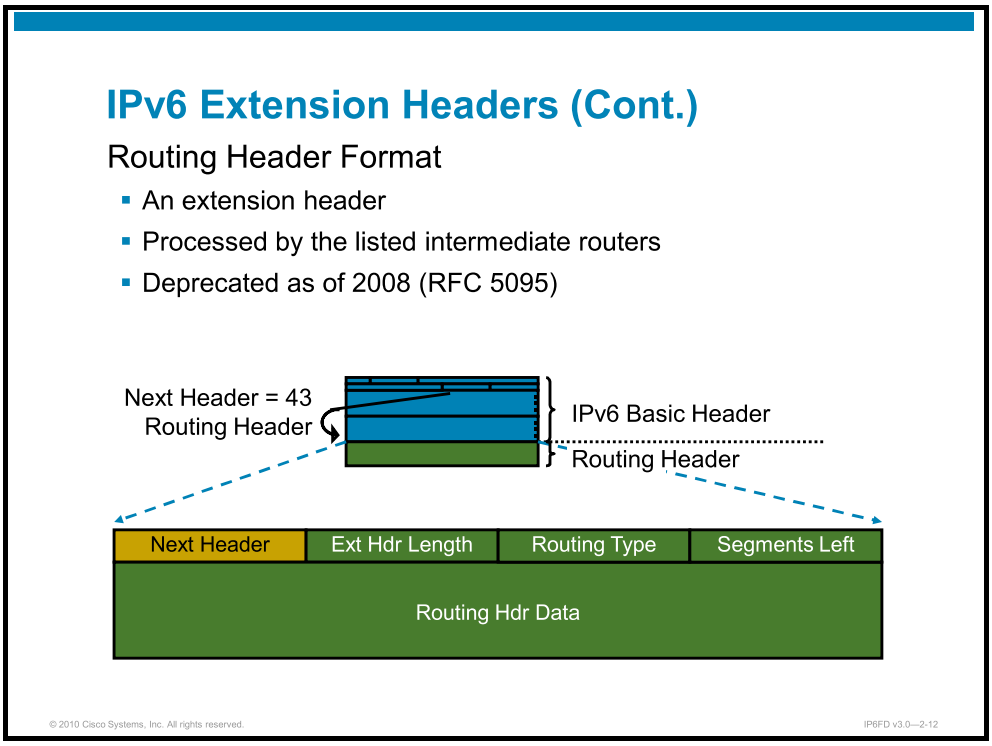


If the Next Header field equals 0, then the next header is a Hop-by-Hop field. This header contains information that must be examined by each node on the path.

The Hop-by-Hop and Destination Option extension headers support one or more options. These options are encoded in the data portion of the extension header, in type, length, value (TLV) format. These extension headers are actually containers for options.

One use for the Hop-by-Hop header is to support IPv6 jumbograms, which are defined in RFC 2675, *IPv6 Jumbograms*. The IPv6 header has a 16-bit Payload Length field, and therefore supports payloads as long as 65,535 octets. The IPv6 Hop-by-Hop option—with the Jumbo Payload option—carries a 32-bit long field. This special option is used to allow transmission of IPv6 packets with payloads that are between 65,536 and 4,294,967,295 octets long.

Tip Partitioning of options into this Hop-by-Hop header and the Destination Options header is a primary feature of IPv6 and helps to improve aggregate throughput on an IPv6 network. By partitioning options in this manner, intermediate routers can make a quick decision about the need to look more deeply into the packet. If the IPv6 base header contains a “0” in the Next Header field, the router must examine the options. If the router sees any other value, it can immediately route the packet, and no further examination is needed. This process helps routers to focus on fast forwarding of packets that do not require special handling along the path.

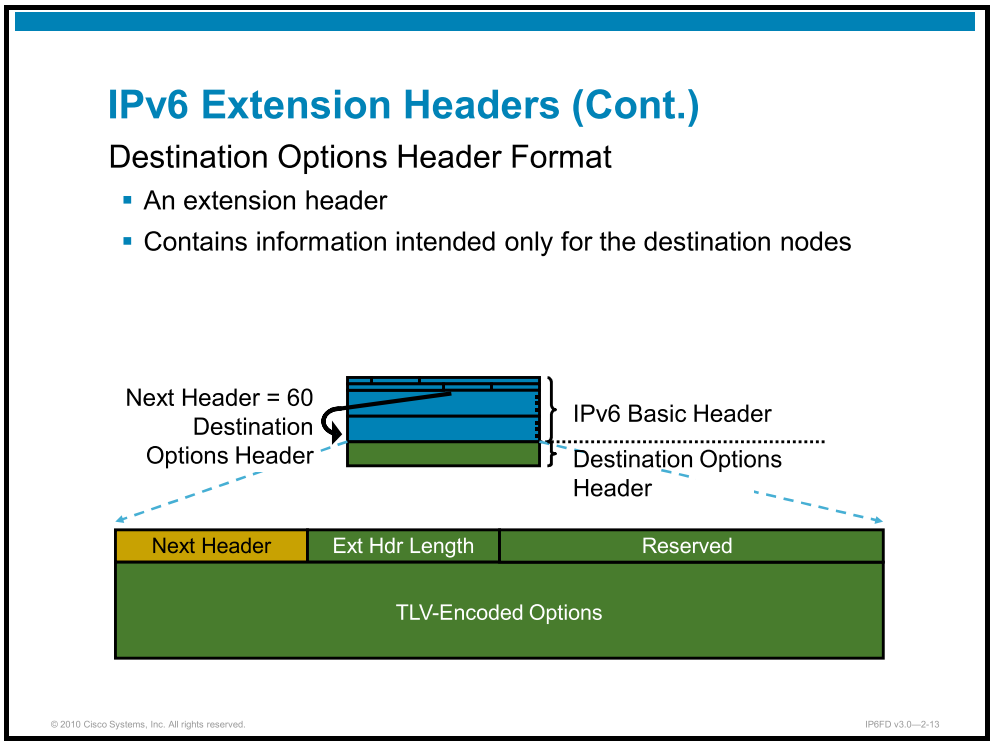


If the Next Header field equals 43, then the next header is a Routing header. A Routing header can appear either as the first extension header after the IPv6 base header, or after another extension header.

As in any extension header, the first field of the Routing header is the Next Header field, which identifies the type of header that follows the Routing header. The second field is the Exterior Header Length field. The Routing Type field identifies the type of Routing header that is used. The Segments Left field identifies the number of intermediate routers that are in the data portion of the Routing header.

A Routing header with routing type 0 forces the routing through a list of intermediate routers, like the Loose Source Route option does in IPv4.

A second Routing header, with routing type 2, has been defined for use with IPv6 Mobility. This header is formatted like the type 0 Routing header but carries only one intermediate hop.

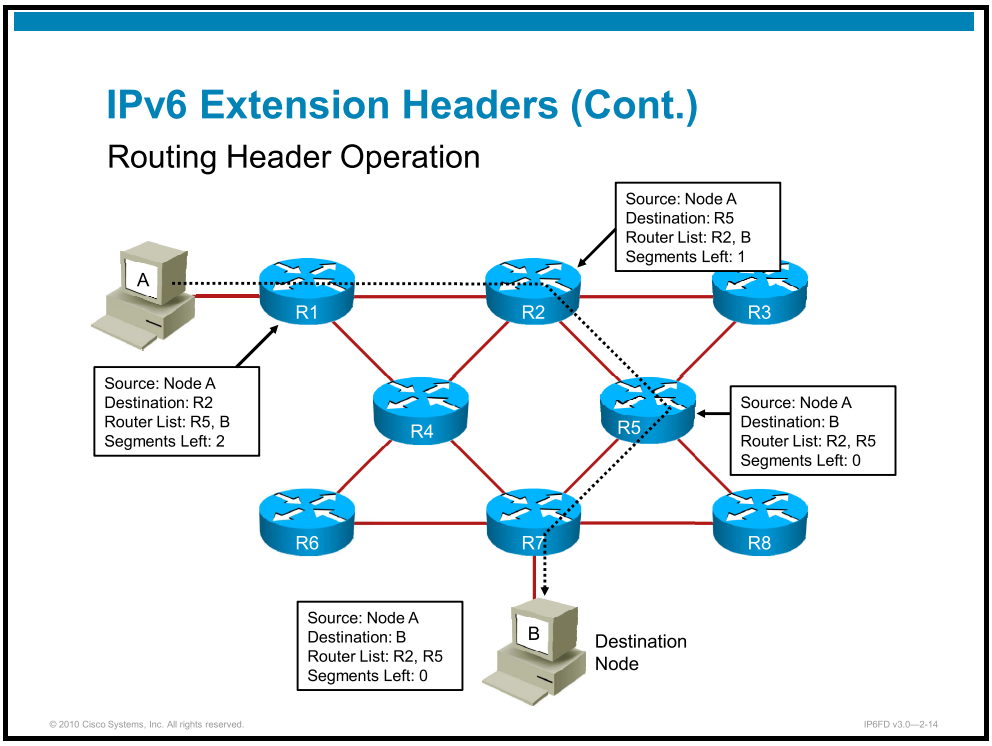


If the Next Header field equals 60, then the next header is a Destination Options header.

An example of a Destination Options header is the Home Address option in Mobile IPv6. The Home Address option is carried by the Destination Options extension header. This option is used in a packet that a mobile node sends while away from home, to inform the recipient of the home address of the mobile node.

This is the only type of Routing header that can appear twice in an IPv6 packet.

Tip Destination options are end to end, which helps to secure sessions because these options can be covered by IPsec protections. The options need not be visible (and are never changed) along the path. Hop-by-hop options are not covered by end-to-end security tools such as IPsec, because they must be visible to intermediate nodes along the path.



The figure shows the changes in the Routing headers and the destination address during the routing of the packet in the path from A to B.

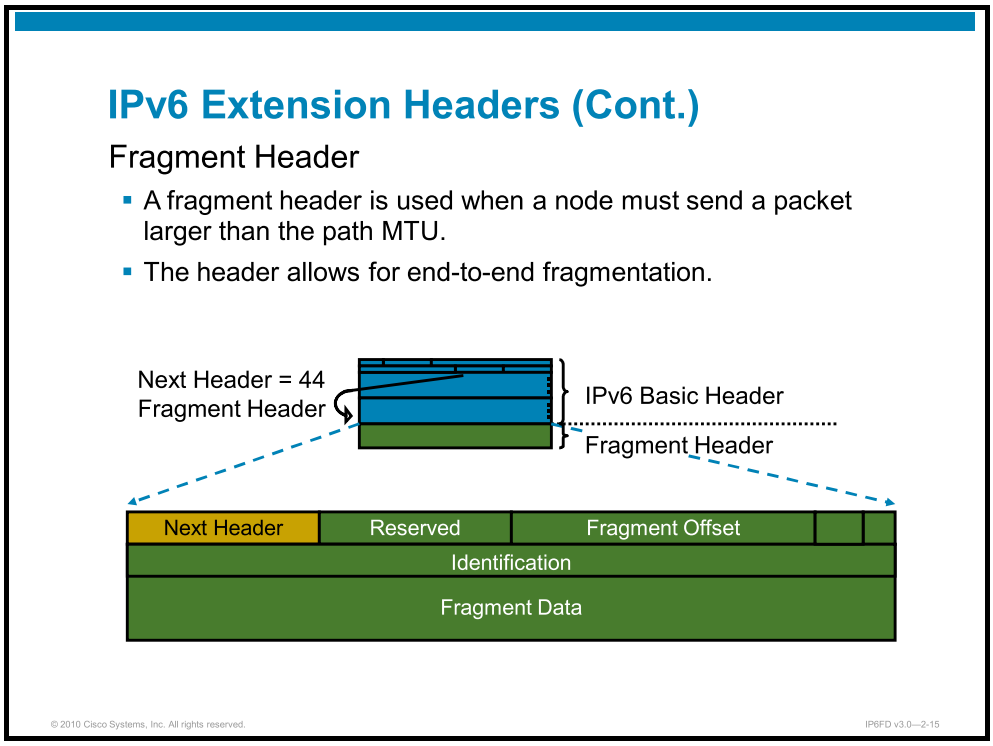
The way in which the Routing header and the destination address in the IPv6 packet interact is new. At each intermediate router in the list, the router changes the destination address of the outgoing packet to target the next-listed router. This list of hops is always saved in the Routing headers.

The number of Routing headers does not change, but the content of the router address inside each Routing header and the packet destination address changes during the path. When the last router in the list receives the packet, the router changes the destination address to the final destination address, which is the address of host B.

Initially, host A puts in an R2 address as the destination address of the packet. Host A also puts in the Routing header the address of R5 (the next router on the list) and the address of host B (the final destination). When the packet leaves R2, R2 changes the destination address to R5 and puts its own address (R2) in the first Routing header. The Segments Left field is decremented by one. When the packet leaves R5, R5 changes the destination address to B, the final destination, and then puts its own address (R5) in the last Routing header.

When the packet arrives at B, the source (A) and the destination address (B) are as if no Routing header were present. However, B can look at the Routing headers to see the path (R2, R5) that the packet took. The exchanging of those addresses in the header does not involve any checksum recalculation.

Putting the next hop in the destination address enables the routing of the packet between the two hops to be processed without any change. For example, between any two hops in the list, the basic routing is based on the longest match routing algorithm and corresponds to the default method for routing packets in IPv6.



The fragment header is used when a node must send a packet that is larger than the path maximum transmission unit (MTU). The path MTU is the largest packet that can be conveyed across a given network path. For example, if each link in the path can carry a 2000-byte packet—except for one link, which can carry only a 1500-byte packet—then the path MTU is 1500 bytes. If the source node has a 1600-byte packet to send, it needs to fragment the packet into two packets.

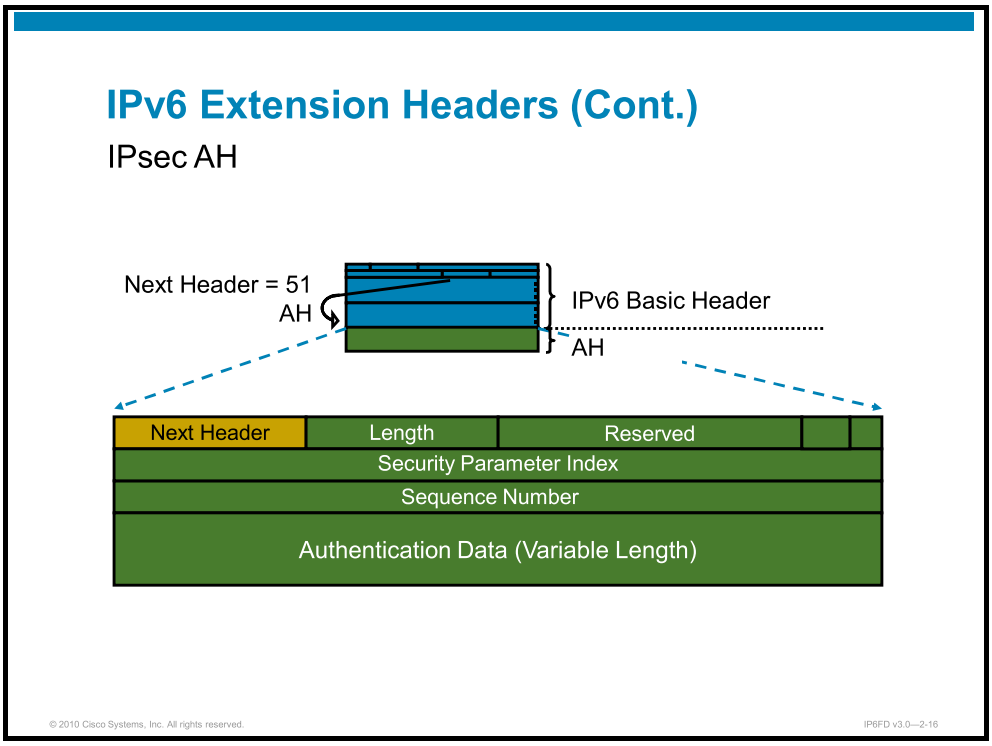
When a packet exceeds the MTU, the source node cuts the packet into fragments and sends each fragment in a separate packet, identifying each fragment by adding the Fragmentation extension header behind the base IPv6 header of each new, smaller packet.

The fields of the Fragmentation header look like the Fragment fields in the IPv4 header, and include the following:

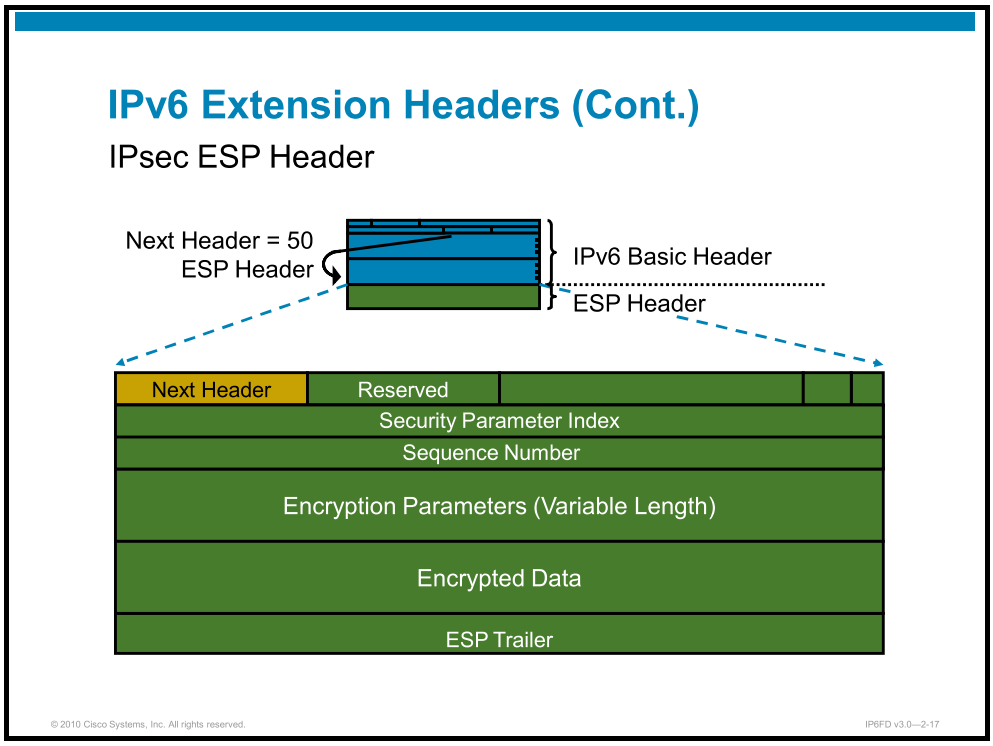
- A fragment offset that identifies the position of the specific fragment in the complete original packet
- An identification number that identifies fragments that are from the same original packet

The destination node then reassembles the packet by concatenating the received fragments in the order that the fragment offset provides.

Note Unlike in IPv4, only the source node in IPv6 is allowed to fragment packets. Routers in the transit path are not allowed to fragment packets. Instead, the router must send an Internet Control Message Protocol version 6 (ICMPv6) error message back to the source, notifying the source that fragmentation is required.



The AH, described in RFC 4302, is an IPsec header that provides packet authentication and integrity checking.



The ESP header, described in RFC 4303, provides confidentiality and data-integrity services for peer nodes.

Tip IPsec ESP can be used with null encryption, which means that only the authentication, integrity, and anti-replay features of ESP are enabled. There is a discussion in the security community about whether IPsec ESP with null encryption is sufficient for all nonprivacy-related sessions, and whether IPsec AH should be eliminated. Note that transport-mode IPsec AH provides broader coverage of packet integrity than ESP does.

IPv6 Extension Headers (Cont.)

Upper-Layer Headers

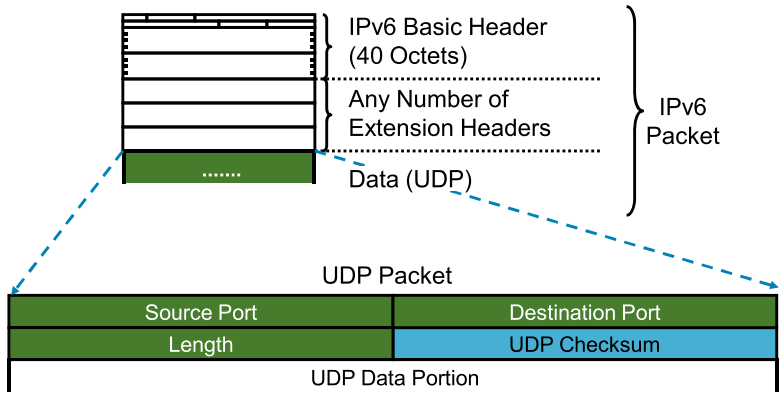
- Used for the payload protocols
  - ICMP
  - UDP
  - TCP
  - SCTP
- Must be the last extension header

Upper-Layer headers in IPv6 are structured the same way as in IPv4. Upper-Layer headers must be last in a chain of extension headers.

IPv6 Extension Headers (Cont.)

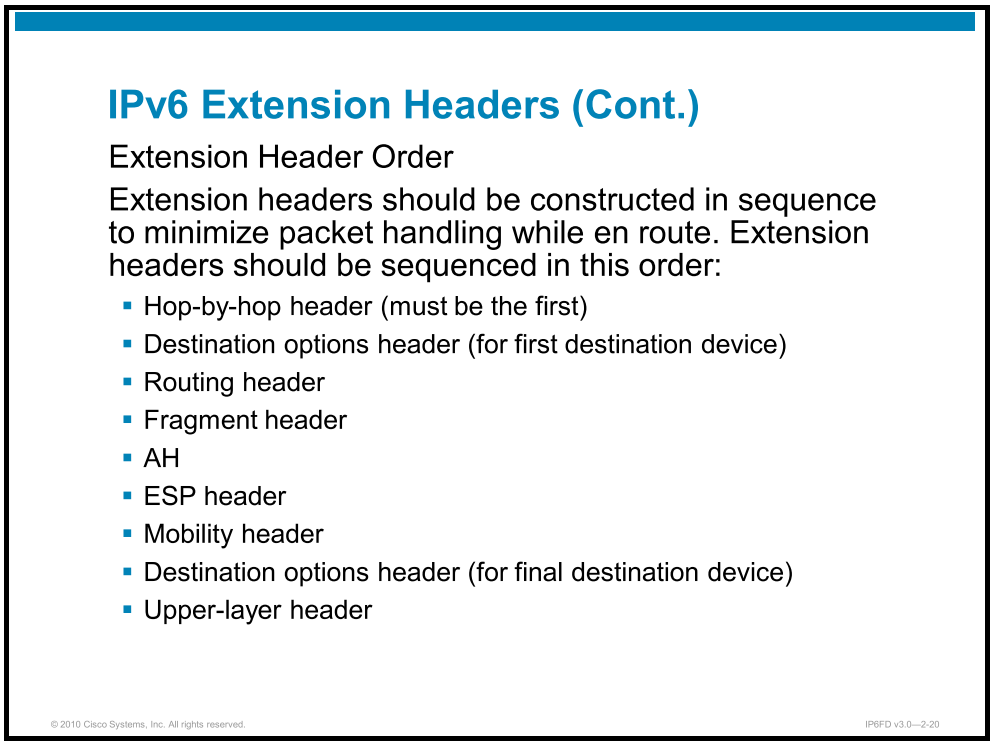
UDP

- Computation of UDP checksum is mandatory for UDP running over IPv6.



In IPv4, the UDP transport layer uses an optional checksum. Because the IP header checksum is removed in IPv6, IPv6 uses the UDP checksum to check the integrity of the inner packet. The checksum is mandatory.

The UDP pseudoheader includes the IPv6 source and destination addresses. If source routing is included, the destination address is computed from the final destination in the source-route path.



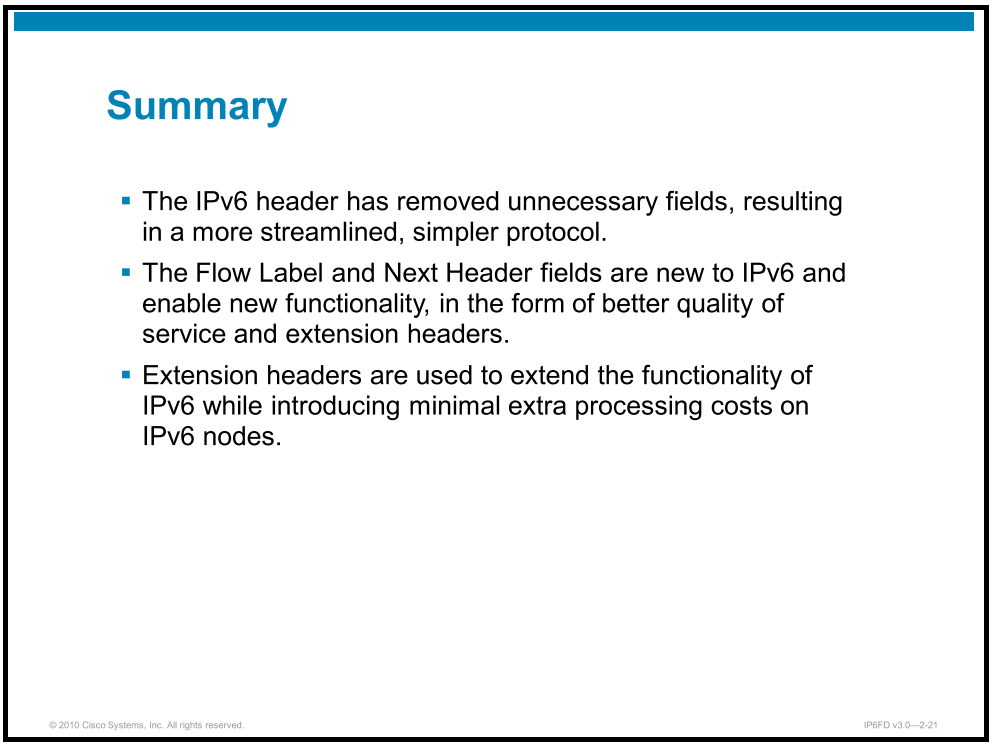
A review of some of the main characteristics of extension headers follows:

- The Hop-by-Hop Options header is the only type of header that is read and processed by all hops in the path. The Hop-by-Hop Options header is used currently for the router alert and jumbogram functions and will be used to carry new options as they are invented in the future.
- The Destination Options header is processed only by the destination node. Currently, only Mobile IPv6 uses the Destination Options header.
- The AH and the ESP headers are used within IPsec to provide authentication, integrity, and confidentiality of the IP packet, and are identical in both IPv4 and IPv6.
- Upper-Layer (transport) headers are the typical headers that are used inside the IP packet to transport the application data. The two main transport protocols are TCP and UDP.

Tip IPv6 engineers sometimes call these headers upper-layer extension headers, which is incorrect terminology. IPv6 extension headers are Layer 3 headers and are part of the IPv6 protocol (including ICMPv6). TCP and UDP, for example, are Layer 4 protocols and are not IPv6 extension headers.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 3

Enabling IPv6 on Hosts

Overview

Support on the end nodes is crucial for implementation of IP version 6 (IPv6). Most, if not all, major operating systems already support IPv6, and there should be no problem implementing IPv6 support on hosts. Application support is a different matter and can vary from application to application. Commercial applications should offer IPv6 support, whereas internally developed applications depend on the internal development team. This lesson describes how to configure basic IPv6 parameters on Microsoft Windows, Mac OS, and Linux kernel-based operating systems.

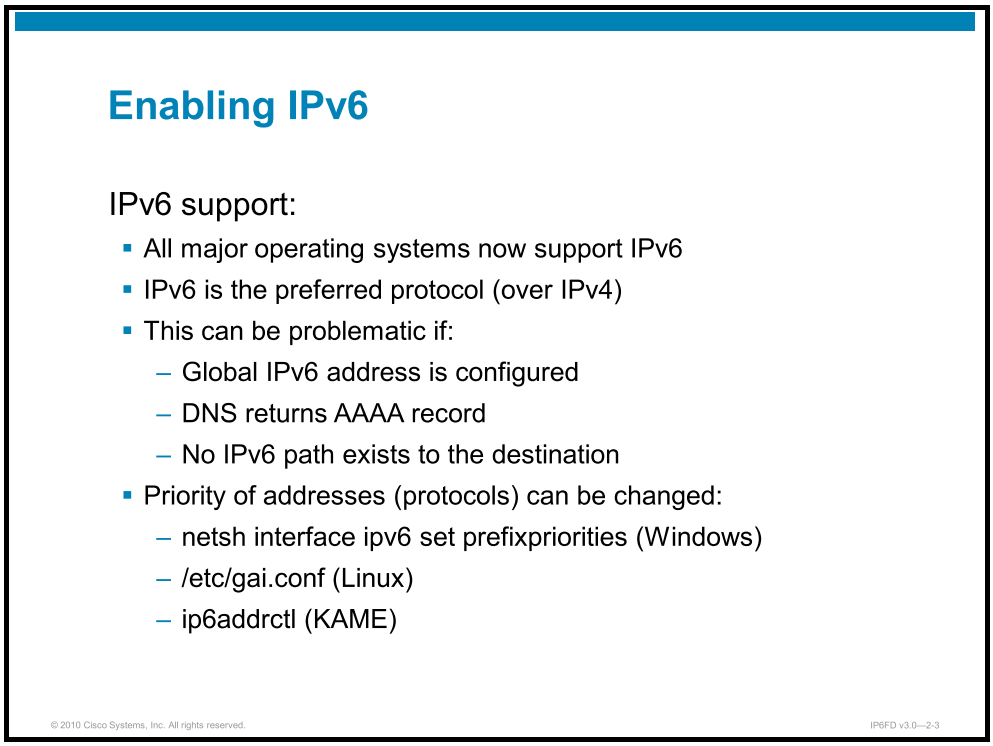
Objectives

Upon completing this lesson, you will be able to configure IPv6 on Windows and Linux-based operating systems. This ability includes being able to meet these objectives:

- Describe how IPv6 is enabled on the hosts
- Describe how IPv6 is enabled on Windows
- Describe how IPv6 is enabled on Mac OS X
- Describe how IPv6 is enabled on Linux-based operating systems

Enabling IPv6 on Hosts

This topic describes IPv6 on end nodes.

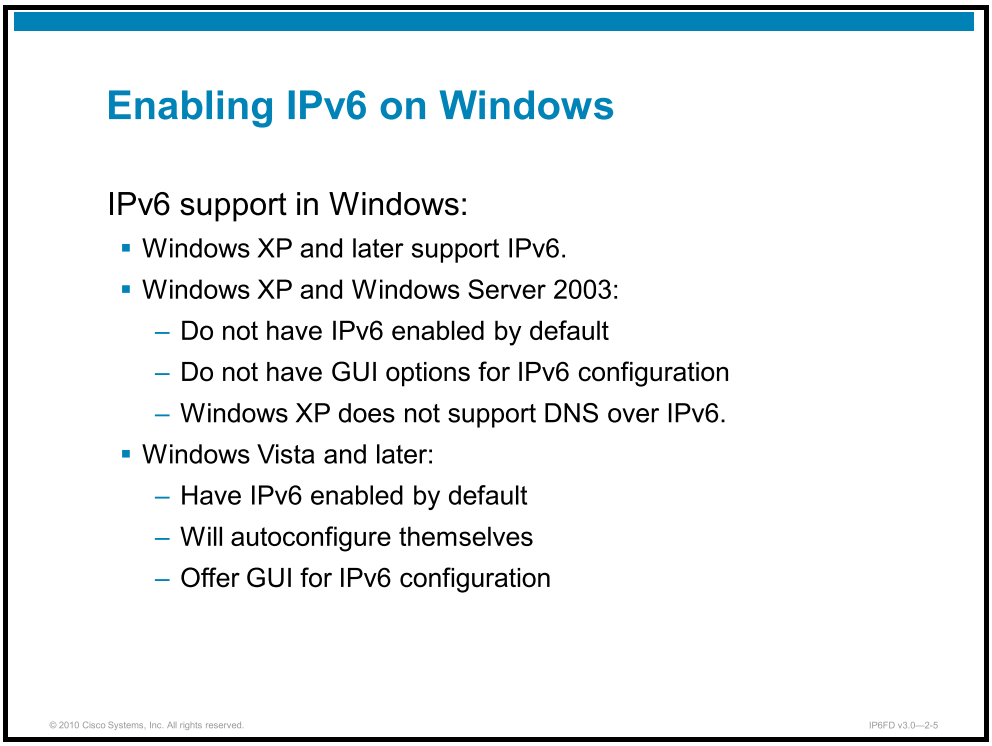


IPv6 must be enabled on end hosts as well as on the entire network. Before enabling IPv6 on end hosts, you should be aware of the potential problems that doing so might create. IPv6 is preferred over IP version 4 (IPv4) on most platforms; if your end host has an IPv6 global address and an IPv6 default route or specific route to the destination, the host probably will try to use IPv6 to reach the destination. If the IPv6 path to the destination is interrupted somewhere—for example, if you have local IPv6 connectivity and advertise the default route inside your network, but your IPv6 Internet connectivity is down—then your hosts will experience apparent slowdown of Internet traffic towards dual-stacked Internet hosts. This issue occurs because of IPv6 preference. To change this behavior, specify higher priority for IPv4 traffic, by increasing precedence value.

On Windows, you can increase this value by using the **netsh interface ipv6 set prefixpriorities** command. On Linux-based operating systems, you can edit the /etc/gai.conf file. On operating systems that have a KAME-based IPv6 stack (Berkeley Software Distribution [BSD] series and Mac OS X), you can use the **ip6addrctl** command.

Enabling IPv6 on Windows

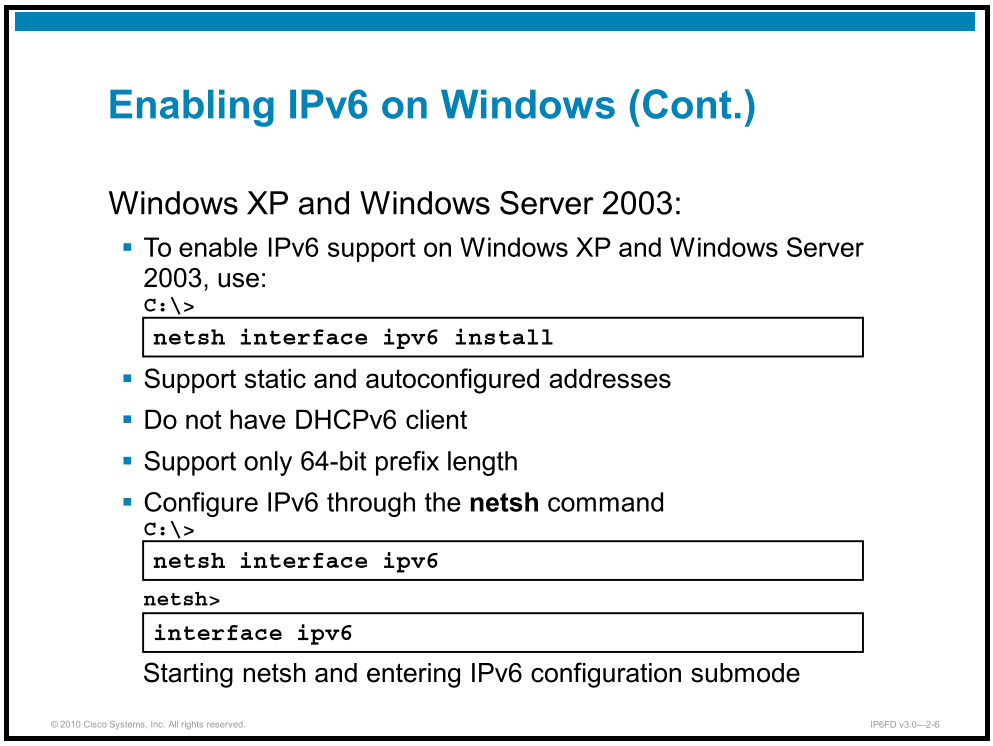
This topic describes how to configure IPv6 on Windows operating systems.



All current versions of the Windows operating system support IPv6. Windows XP and Windows Server 2003 have limited configuration options for IPv6. There are no GUI options, except the ability to install support for IPv6. Configuration is done through the netsh command-line interface (CLI).

Windows Vista and later editions have complete GUI support for configuring IPv6, as well as the netsh CLI.

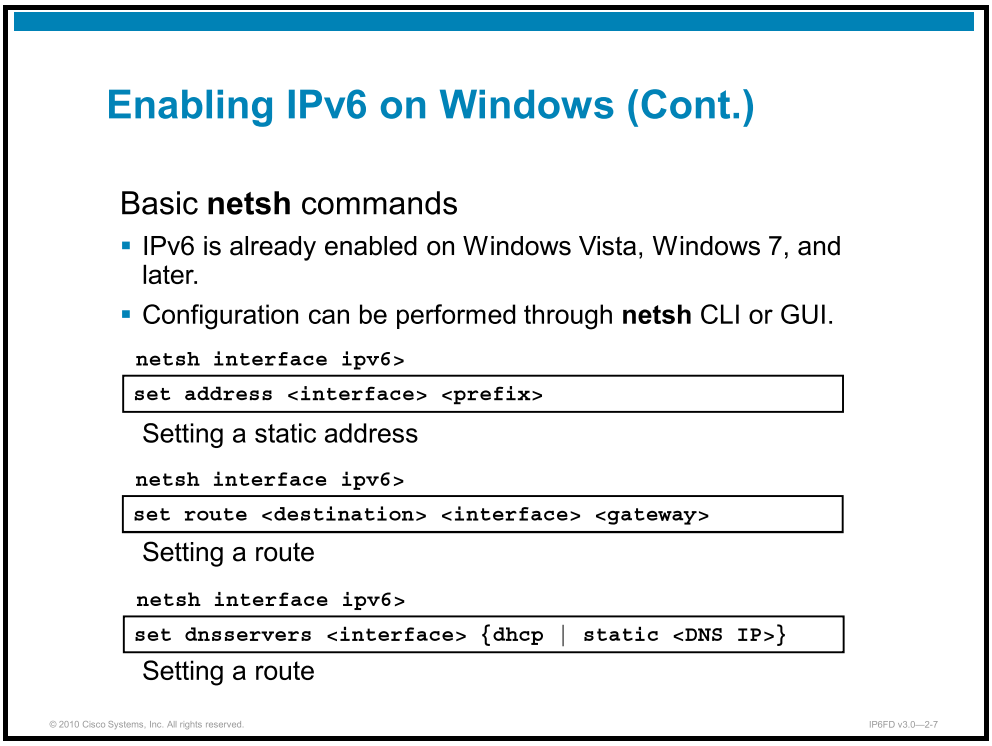
By default, all versions autoconfigure themselves with global IPv6 addresses as soon as a router on the segment advertises a prefix via neighbor discovery router advertisement.



On Windows XP and Windows Server 2003, IPv6 is available but is not enabled by default. To activate support for IPv6, you must enter the **netsh interface ipv6 install** command once. The change is permanent and persistent across reboots.

After enabling IPv6 (no reboot is required), you can leave the PC to autoconfigure itself or you can set a static IPv6 address. However, Windows XP and Windows Server 2003 do not have a DHCP version 6 (DHCPv6) client, do not support an arbitrary prefix length (only 64-bit prefixes are supported) and do not support DNS transport over IPv6.

When you configure through the netsh CLI, the easiest way is to start netsh by using the **netsh** command, then enter IPv6 configuration submode by entering **interface ipv6** and pressing the Enter key.



On Windows Vista and later, IPv6 is enabled by default. A fresh installation will automatically compute the IPv6 address if a router is present and sending route advertisement packets.

Configuration can be done either through the CLI or GUI. CLI commands are the same as in previous releases of Windows.

All the commands in this lesson are configured under the interface IPv6 subconfiguration mode. You can execute the commands by specifying each command individually at the cmd.exe prompt, as in this example:

```
c:\>netsh interface ipv6 set address LAN 2001:db8:100::1
```

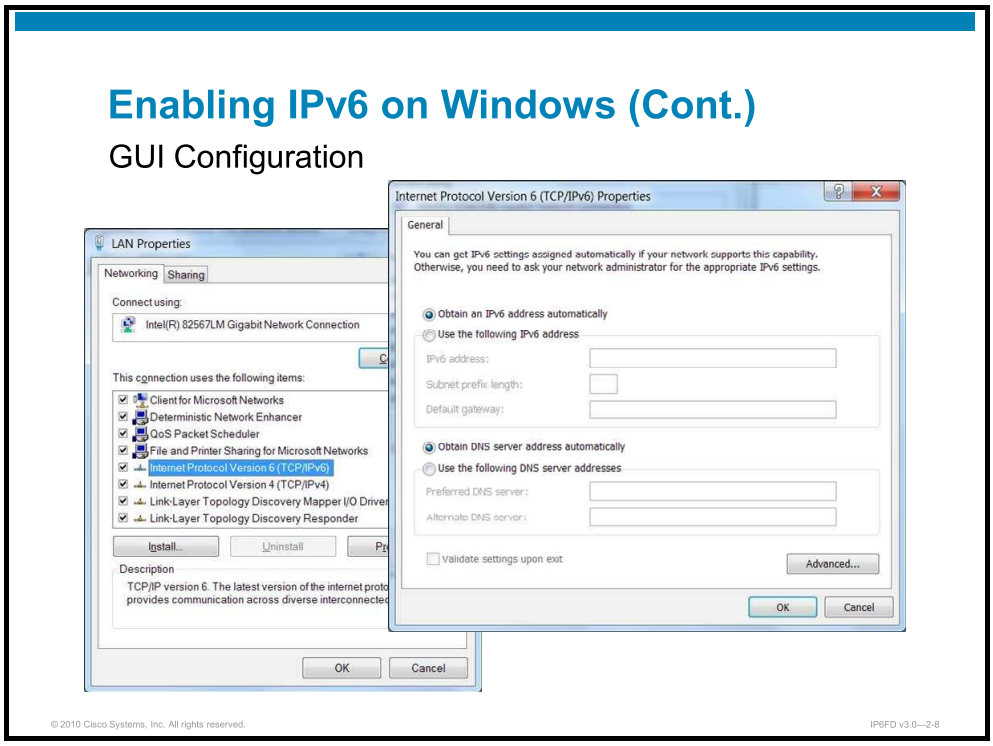
Or you can enter the netsh application, select the configuration mode, and then enter the commands:

```
c:\>netsh
netsh>interface
netsh interface>ipv6
netsh interface ipv6>set address LAN 2001:db8:100:40::1
```

The **set address** ***interface prefix*** command sets a static IPv6 address on an interface.

To configure a route, use the **set route** ***destination gateway*** command. For example, to configure a default route, use the **set route ::/0 LAN fe80::1** command.

DNS servers can be either acquired through DHCP-lite or configured statically. To tell the PC to acquire DNS server information via DHCP-lite, use the **set dnsserver LAN dhcp** command. To statically configure a DNS server use the **set dnsserver LAN static 2001:db8:100:1::53** command.



GUI configuration is very like IPv4 configuration. The figure shows a sample empty configuration of the IPv6 protocol. An IPv6 address can be configured statically or acquired automatically. When configuring static address, you need to specify the IP address, prefix length, and default gateway. There is no more netmask setting; the prefix length is used instead.

DNS servers are still listed. For any additional DNS servers or other settings, click the Advanced button.

Enabling IPv6 on Windows (Cont.)

Verification of Configuration

```
C:\> ipconfig
Ethernet adapter LAN:

   Connection-specific DNS Suffix  . : pc.example.com
   IPv6 Address. . . . . . . . . . . : 2001:db8:1:10:4860:1624:5c13:4512
   Temporary IPv6 Address. . . . . . : 2001:db8:1:10:dc6c:3a19:b210:8632
   Link-local IPv6 Address . . . . . : fe80::4860:1624:5c13:4512%15
   IPv4 Address. . . . . . . . . . . : 192.0.2.116
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::214:50ff:fee0:1627%15
                                       192.0.2.65
```

```
C:\> netsh interface ipv6 show addresses
Interface 15: LAN

Addr Type  DAD State   Valid Life Pref. Life Address
---------  ----------- ---------- ---------- ------------------------
Public     Preferred  29d23h59m37s 6d23h59m37s
2001:db8:1:10:4860:1624:5c13:4512
Temporary  Preferred   2d1h56m28s 2d1h56m28s
2001:db8:1:10:dc6c:3a19:b210:8632
Other      Preferred     infinite   infinite fe80::214:50ff:fee0:1627%15
```


When verifying configuration, you can use either traditional commands or **netsh** commands. For checking the configured address, you can use either the **ipconfig** or **netsh interface ipv6 show addresses** command. The **ipconfig** command shows both IPv4 and IPv6 configurations but offers less information. The **netsh** command is limited to IPv6 configurations unless you modify the command, but this command shows additional statuses such as address type, Duplicate Address Detection (DAD) state, and valid and preferred lifetime.

Enabling IPv6 on Windows (Cont.)

Verification of Configuration (Cont.)

```
C:\> route print -6
===========================================================================
Interface List
 15...00 50 b2 56 13 ed ......LAN Adapter
===========================================================================

IPv6 Route Table
===========================================================================
Active Routes:
 If Metric Network Destination      Gateway
 15    276 ::/0                     fe80::213:60ff:fee0:f126
 15     28 2001:db8:1:10::/64       On-link
 15    276 2001:db8:1:10:4860:1624:5c13:4512/128
                                    On-link
 15    276 2001:db8:1:10:dc6c:3a19:b210:8632/128
                                    On-link
 15    276 fe80::/64                On-link
 15    276 fe80::214:50ff:fee0:1627/128
                                    On-link
 15    276 ff00::/8                 On-link
===========================================================================
Persistent Routes:
  None
```


To check the routing table or the default gateway (more common on end hosts), use the traditional **route print -6** command.

Enabling IPv6 on Windows (Cont.)

Verification of Configuration (Cont.)

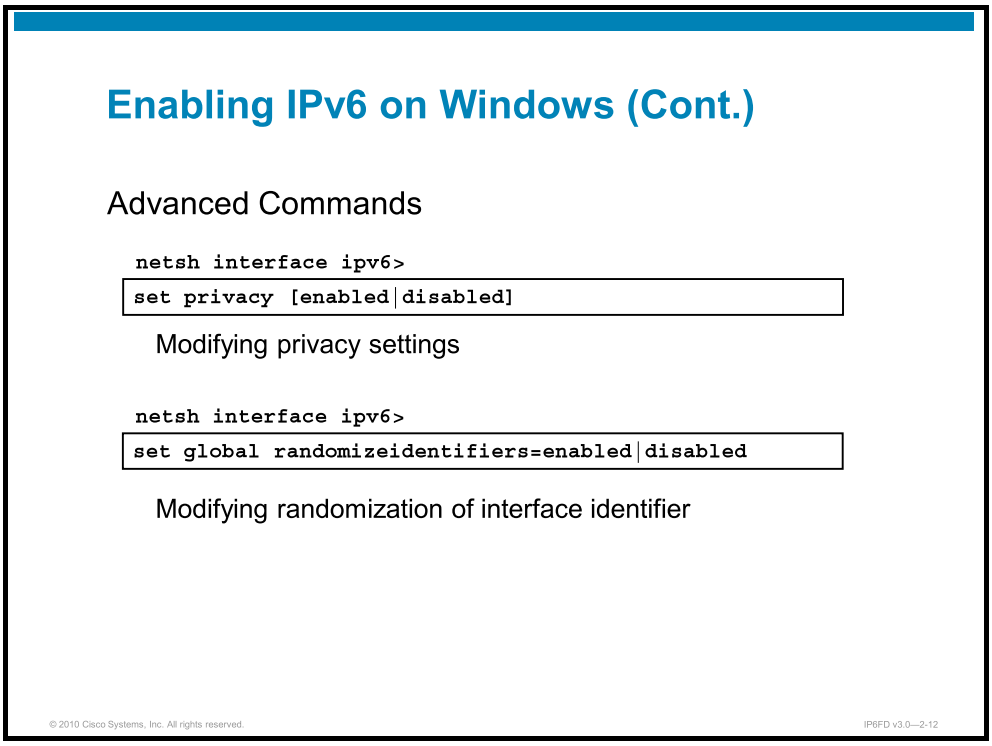
```
C:\> netsh interface ipv6 show route

Publish  Type      Met  Prefix                    Idx  Gateway/Interface Name
-------  --------  ---  ------------------------  ---  ------------------------
No       Manual    256  ::/0                       15  fe80::213:60ff:fee0:f126
No       Manual    8    2001:db8:1:10::/64         15  LAN
No       Manual    256  2001:db8:1:10:4860:1624:5c13:4512/128   15  LAN
No       Manual    256  2001:db8:1:10:dc6c:3a19:b210:8632/128   15  LAN
No       Manual    256  fe80::/64                  15  LAN
No       Manual    256  fe80::214:50ff:fee0:1627/128   15  LAN
No       Manual    256  ff00::/8                   15  LAN
```

- DNS configuration can be checked with:
  - **netsh interface ipv6 show dnsservers**
  - **ipconfig /all**
- Neighbors can be viewed with:
  - **netsh interface ipv6 show neighbors**

An alternative way to check the routing table is to use the **netsh interface ipv6 show route** command.

DNS configuration can also be checked by using the **ipconfig /all** command or the **netsh interface ipv6 show dnsservers** command.

To view a neighbor cache, you must use the **netsh interface ipv6 show neighbors** command.



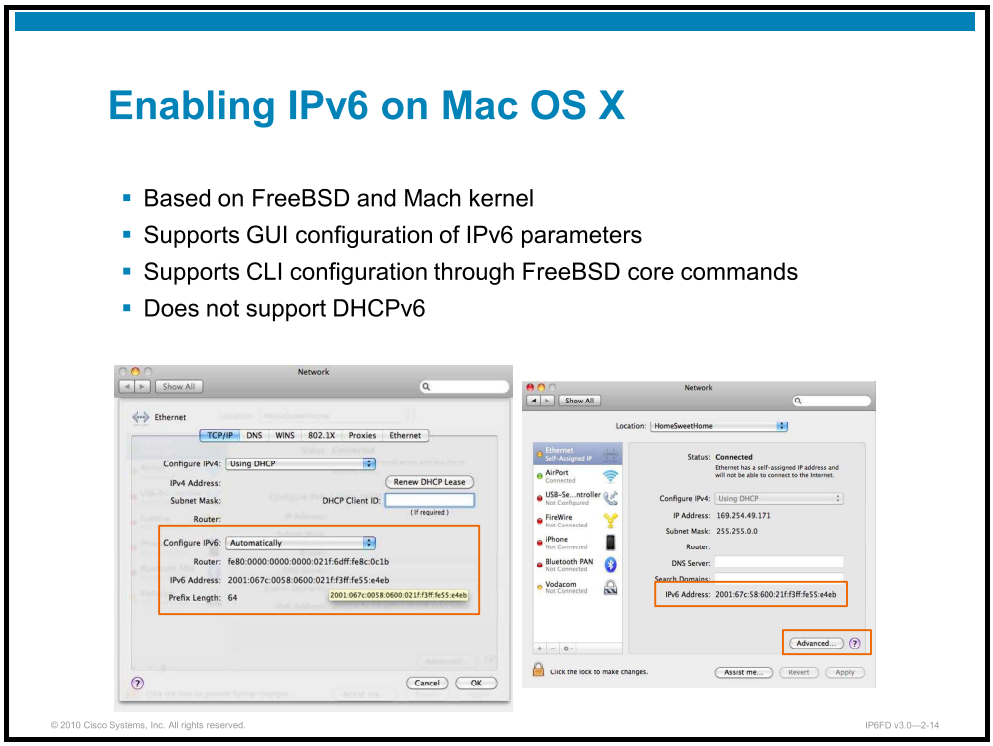
The advanced configuration options include the **netsh interface ipv6 set privacy** command. This command adds an additional, temporary address on every IPv6 interface. Windows uses this address when choosing the source address for any outgoing connections. Additionally, no services listen on this address. Use of this address can prevent a malicious user from successfully port-scanning the PC, because all services listen only on the primary, not on the temporary, address. This feature is enabled by default.

The **netsh interface ipv6 set global randomizeidentifiers** command causes the interface identifier portion of the autoconfigured IPv6 address to be computed by using a pseudo-random number generating algorithm. This feature is beneficial in two ways:

- Determining the hardware manufacturer of the network interface card (NIC) just by looking at the IP address becomes impossible.
- Guessing the node address, by guessing the vendor and assuming the extended universal identifier (EUI)-64 conversion of the MAC address into the interface identifier, becomes less probable. If an attacker guesses the vendor it reduces address space to /24, which is still 16 million possible addresses but is an order of magnitude smaller than /64.

Enabling IPv6 on Mac OS X

This topic describes basic commands to configure IPv6 on Mac OS X.



Mac OS X is the current Apple operating system for Apple hardware. Mac OS X is based on the Mach microkernel, FreeBSD core utilities, and a GUI that draws from NeXTstep technologies, which was developed for NeXT computer systems.

Mac OS X natively supports IPv6. Configuration is possible through the GUI or through CLI commands. CLI commands for Mac OS X are like GNU or Linux commands:

- **ifconfig:** Is used for manual configuration of IP parameters on interfaces
- **route:** Is used for configuring the routing table
- **netstat:** Shows network status, routing table, and interface statistics

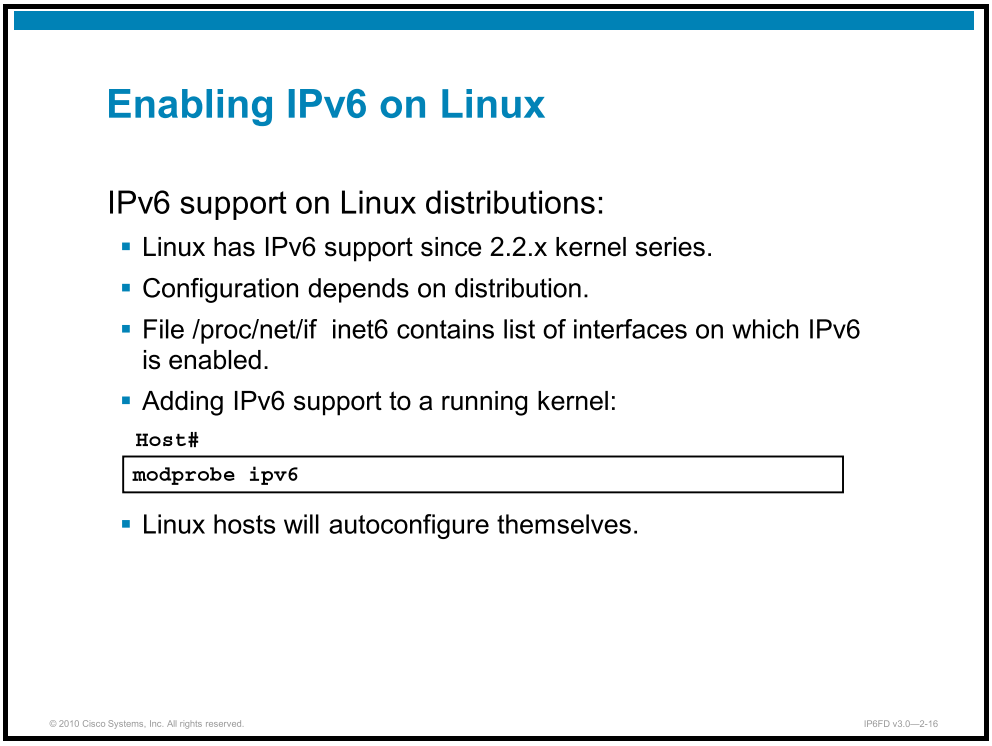
You can also use the **ip6** command to globally enable or disable IPv6:

- **ip6 –a:** Enables IPv6 on all interfaces
- **ip6 –x:** Disables IPv6 on all interfaces

DHCPv6 is not supported natively on Mac OS X. Because the operating system is based on BSD, you should be able to install third-party DHCP products that work on FreeBSD, to provide DHCPv6 functionality on Mac OS X.

Enabling IPv6 on Linux

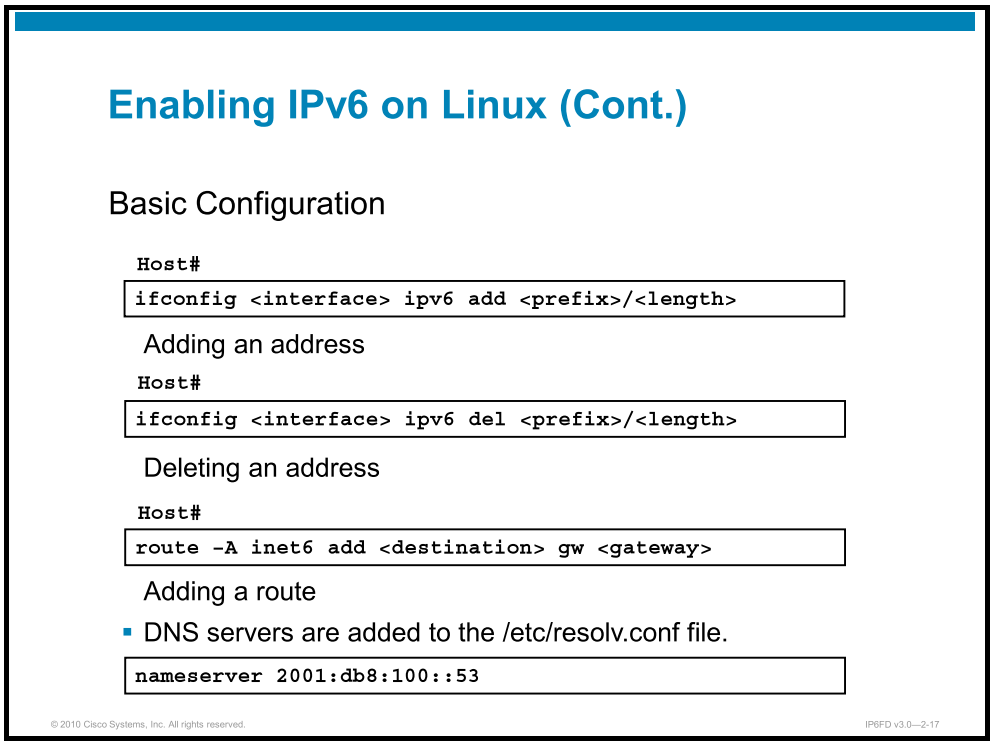
This topic describes the basic commands to configure IPv6 on operating systems that are based on the Linux kernel.



Linux-based operating systems use Linux as the operating system kernel. The Linux kernel supports IPv6 since the kernel version 2.2. Where the configuration is stored depends on the distribution, but basic commands are the same regardless of the version of the kernel. Note that configuring your host as described in this lesson does not preserve settings across reboots.

To verify whether your Linux host supports IPv6, look for the file /proc/net/if inet6. This file should have one line per interface on which IPv6 is enabled. If you do not find this file, try loading the IPv6 kernel module with this command:

- **modprobe ipv6**



If you want to add an address to an interface, you can use the **ifconfig** *interface* **ipv6 add** *prefix/length* command; for example:

- **ifconfig eth0 ipv6 add 2001:db8:100:12::10:17/64**

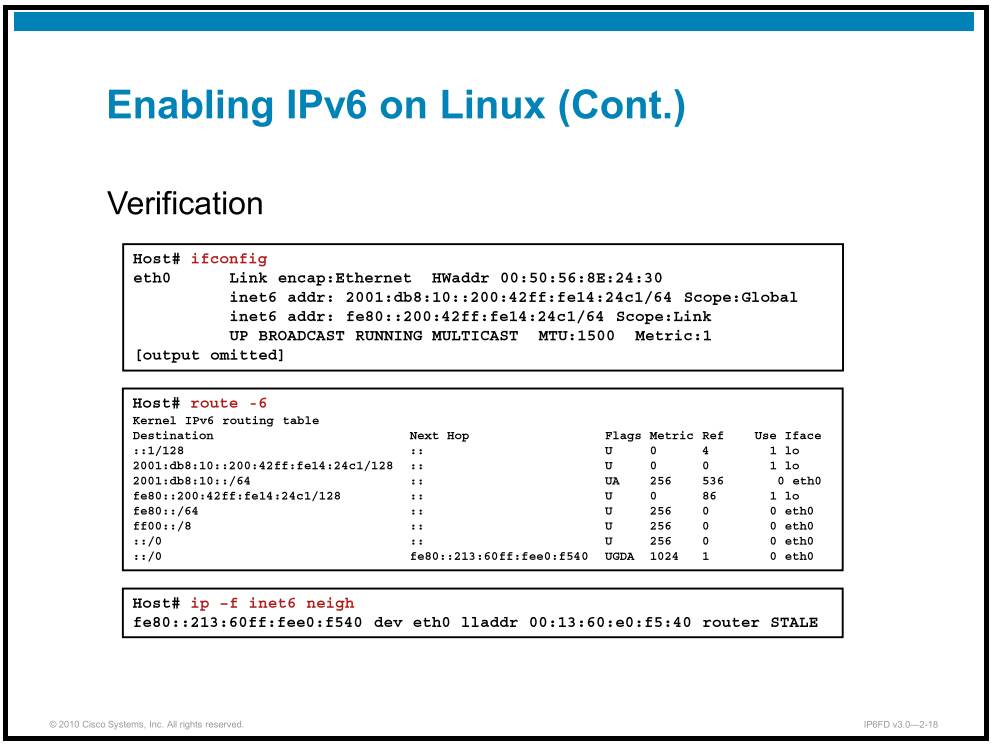
Removing an address is analogous. The **add** keyword is replaced by the **del** keyword.

To add a route, you use the **route –A inet6 add** *destination* **gw** *gateway* command.

The alias **default** can be used to add a default route; for example:

- **route –A inet6 add default gw fe80::1**

DNS servers are added to the /etc/resolv.conf file, one server IP address per line and prefixed with the **nameserver** keyword.



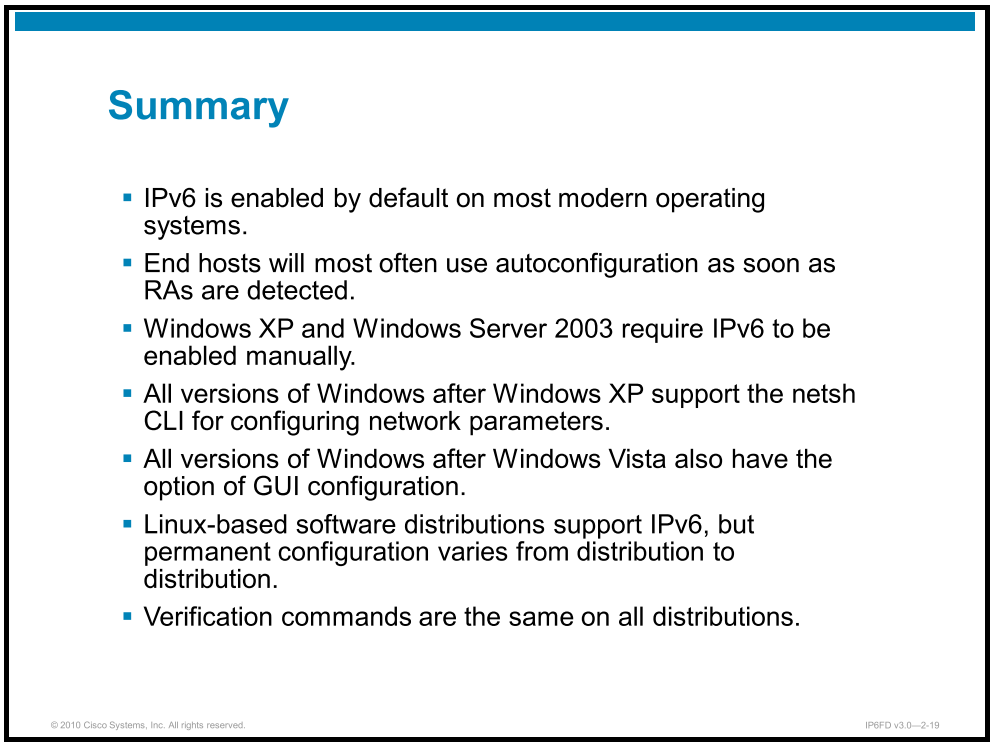
Verification of configured addresses is done by issuing the **ifconfig** command, which displays addresses and some statistics for all the interfaces that are installed on the system.

Checking of IPv6 routes is done by issuing the **route -6** command.

A list of neighbors can be viewed only by using the **ip** command, which is a part of the iproute2 package. This package allows changing and viewing of all other parameters as well. The syntax for listing the neighbors is **ip –f inet6 neigh**.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 4

Enabling IPv6 on Cisco Routers

Overview

Under development since the mid 1990s, IP version 6 (IPv6) can be found in most modern networking hardware and software. Cisco is no exception and has provided support for IPv6 in Cisco IOS Software since 2002. Network engineers and administrators who support IPv6 integration in Cisco Powered Networks require knowledge of the Cisco IOS Software commands that are used to configure and support IPv6 on Cisco routers. This lesson describes IPv6 enablement on Cisco IOS Software for routers.

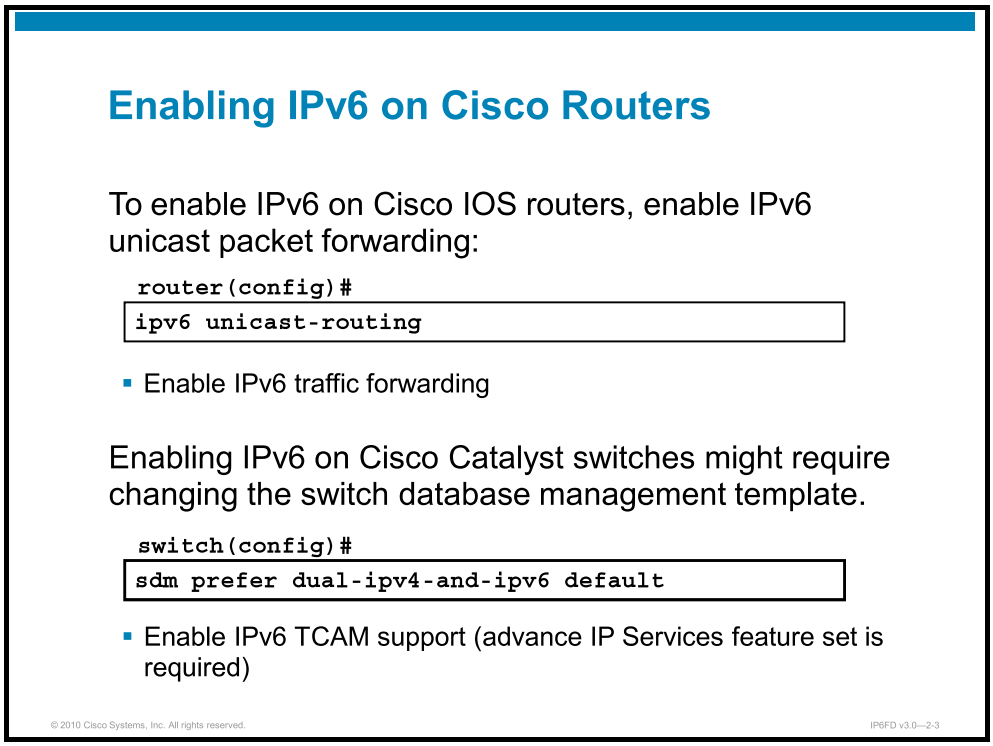
Objectives

Upon completing this lesson, you will be able to describe and use Cisco IOS Software commands to enable IPv6 on Cisco routers. This ability includes being able to meet these objectives:

- Use Cisco IOS Software commands to enable IPv6 on Cisco routers
- Configure IPv6 addresses on Cisco router interfaces
- Explain the two types of IPv6 autoconfiguration

Enabling IPv6 on Cisco Routers

This topic describes how to enable IPv6 forwarding on Cisco routers.



Follow these two basic steps to activate IPv6 on a router:

Step 1 Activate IPv6 traffic forwarding.

Step 2 Configure each interface in which IPv6 is required.

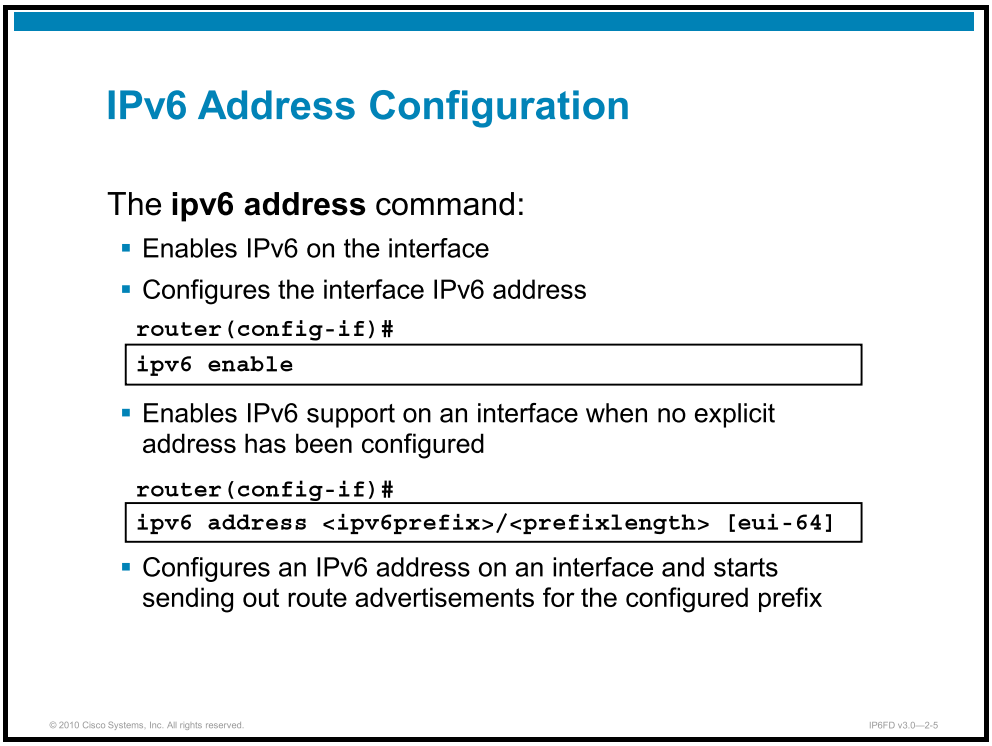
By default, IPv6 traffic forwarding is disabled on a Cisco router. To activate IPv6 traffic forwarding between interfaces, configure the **ipv6 unicast-routing** global command. This command allows the forwarding of unicast IPv6 traffic.

IPv6 is enabled on a per-interface basis. Configuring an IPv6 address on an interface automatically configures the interface link-local address and activates IPv6 for that interface.

On Cisco Catalyst switches, a change of database management template might be required. Use the **sdm prefer** command to change how ternary content addressable memory (TCAM) is divided. To enable IPv6 support, use the **sdm prefer dual-ipv4-and-ipv6 default** command. This command will slice TCAM in a switch so that IPv6 entries can fit into it. The command works only on Layer 3 switches with an advanced IP Services feature set.

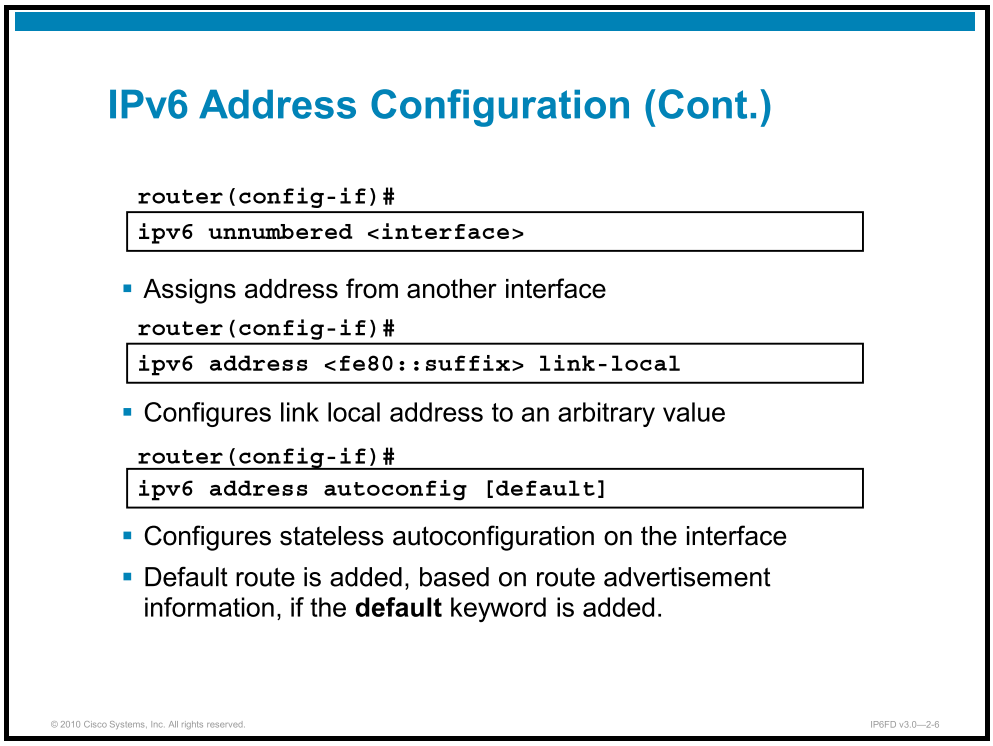
IPv6 Address Configuration

This topic describes how to configure IPv6 addresses on Cisco router interfaces.



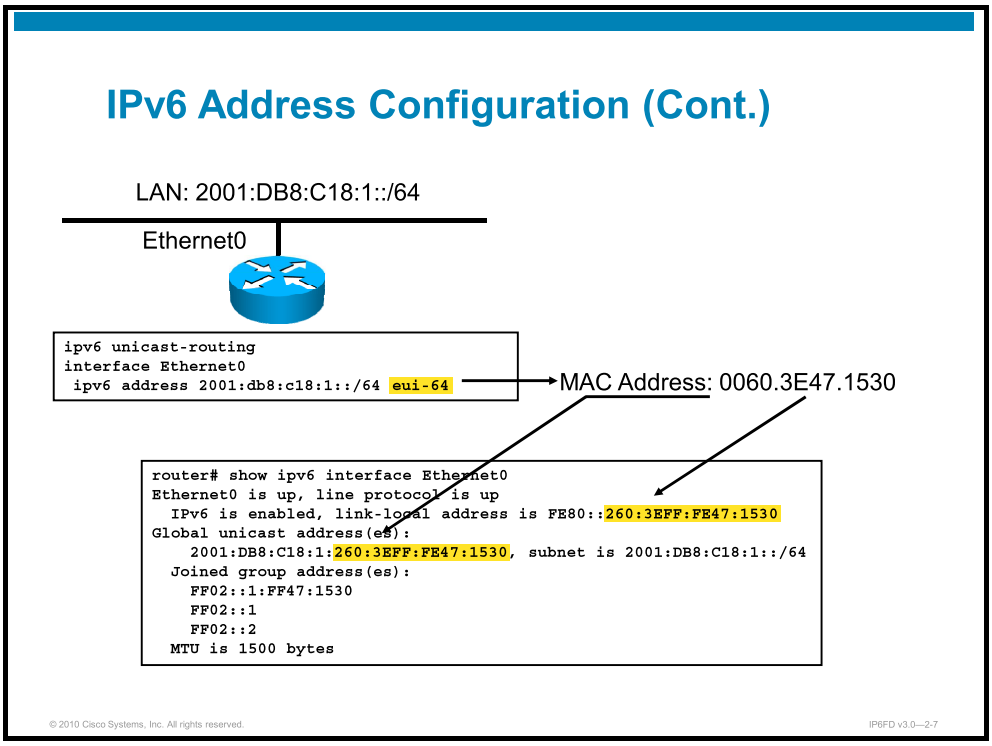
To configure the IPv6 address on an interface, use one of these five commands:

- **ipv6 enable:** You can enable IPv6 on the interface without specifying any IPv6 address. The **ipv6 enable** command enables IPv6 and automatically configures the link-local address for this interface. If no other address is configured, then the interface will have a link-local address only. The link-local address can be used to communicate only with nodes on the same network link (neighbors).
- **ipv6 address <ipv6prefix>/<prefix-length> [eui-64]:** This address command can configure global IPv6 addresses. The link-local address is configured automatically when an address is assigned to the interface. The entire 128-bit IPv6 address must be specified, or a 64-bit prefix must be specified and the **eui-64** option must be used.



- **ipv6 unnumbered <interface>:** You can also configure IPv6 interfaces without explicitly configuring a global IPv6 address for each interface. The **ipv6 unnumbered** *interface* command instructs the unnumbered interface to use the configured global IPv6 address of the specified interface as the source address of the packets that originate from the unnumbered interface.
- **ipv6 address <fe80::ipv6addr> link-local:** You can statically define the link-local address by using the **link-local** option. You do not need to specify the prefix length when using a static link-local address.
- **ipv6 address autoconfig [default]:** This command allows the router to autoconfigure itself, based on router advertisements of another router. The optional **default** keyword also adds a default route that is based on route advertisements.

Note Allowing the router to choose its own 64-bit Interface identifier makes sense if there is no need to reach the router from the subnet; for example, for management purposes. This approach also makes it more difficult for scanning-based attacks to find routers, because most manually configured addresses (such as FE80::1) are easy to guess. A good solution might be to use locally generated interface identifiers for router interfaces but to configure the router with a loopback interface with global scope for management purposes.



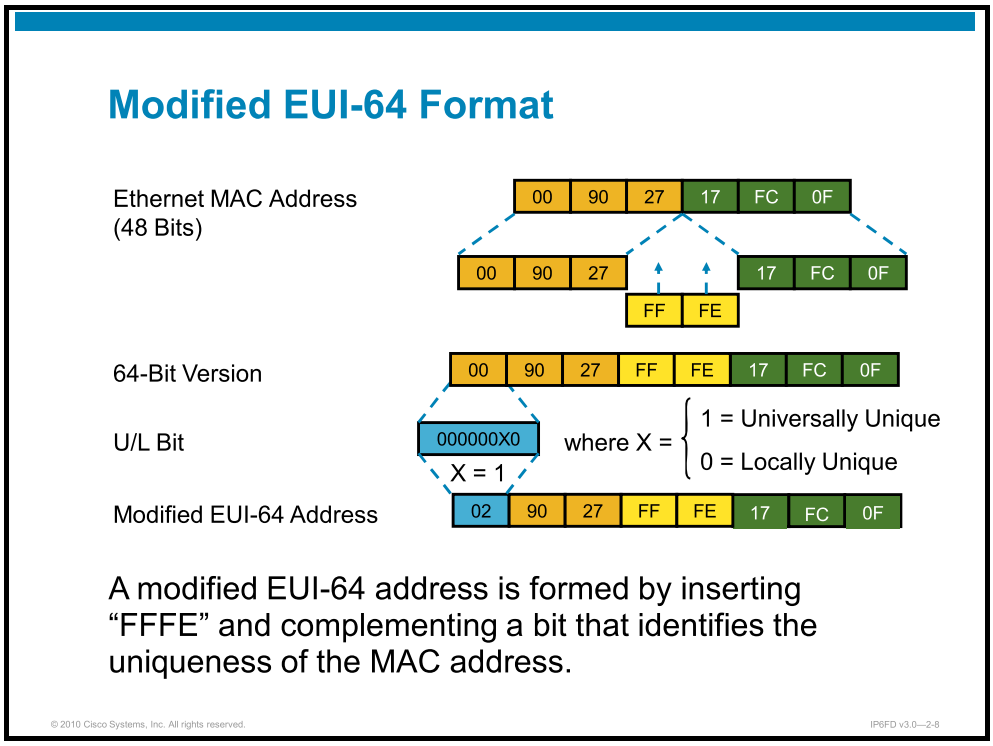
The IPv6 address can be completely specified, or the host identifier (the right-most 64 bits) can be computed from the extended universal identifier (EUI)-64 of the interface. In the example, the IPv6 address of the interface is configured by using the EUI-64 format.

The configuration of the IPv6 address on an interface automatically configures the link-local address for that interface. Also, the interface automatically joins these required multicast groups for that link:

- Solicited-node multicast address FF02::1:FF47:1530
- All hosts on the link multicast addresses FF02::1
- All routers on the link multicast addresses FF02::2

The solicited-node multicast address is used in the Duplicate Address Detection (DAD) algorithm and neighbor discovery.

A solicited-node multicast address is joined for each IPv6 unicast and anycast address that is configured on the interface.



The interface identifier for stateless autoconfiguration in an Ethernet environment uses the modified EUI-64 format. The EUI-64 format expands the 48-bit Ethernet MAC address format to a 64-bit version, by inserting "FFFE" in the middle of the 48 bits.

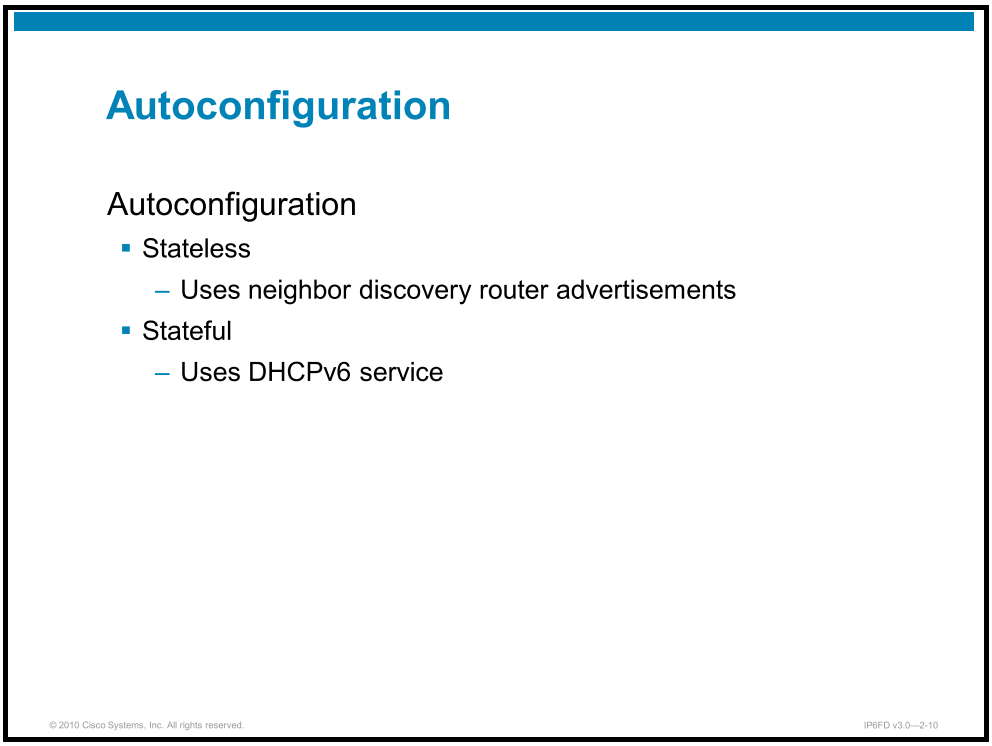
The seventh bit (starting with the left-most bit) in an IPv6 interface identifier is referred to as the Universal/Local (U/L) bit, which identifies whether this interface identifier is universally unique or is locally unique on the link. If the interface identifier was created from an Ethernet MAC address, it is assumed that the MAC address is universally unique and thus, so is the interface identifier.

The U/L bit is for future use of the upper-layer protocols, to uniquely identify a connection, even when there is a change in the left-most part of the address. However, this feature is not yet in use.

Tip Notice that bit 7 is complemented, to mark an EUI-64 address as having been built from a globally unique token. Today, Organizationally Unique Identifiers (OUIs) are assigned with bit 7 set to 0. When bit 7 is complemented, the EUI-64 counterpart always has a 1 in that bit position. This technique makes it easy for you to configure manual addresses, because these bits are left at 0 to indicate local significance. Consider the address 2001:DB8:8:AB::35. Setting bit 7 of the interface identifier to 0 is easier than setting that same bit to 1, which would be 2001:DB8:8:AB:02::35.

Autoconfiguration

This topic describes the two types of IPv6 autoconfiguration.

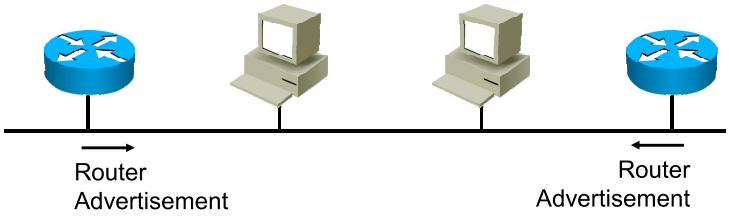


Autoconfiguration is an integral component of IPv6. The two types of IPv6 autoconfiguration are as follows:

- **Stateless autoconfiguration:** Uses neighbor discovery mechanisms to find routers and dynamically create IPv6 addresses
- **Stateful autoconfiguration:** Uses a DHCP version 6 (DHCPv6) server to assign IPv6 addresses to nodes

Router Advertisements

Routers send periodic router advertisements to the all-nodes multicast address.



Router advertisement packet:

- ICMP type 134
- Source = Router link-local address
- Destination = FF02::1 (all-nodes multicast address)
- Data = Options, prefix, lifetime, autoconfiguration flag

Router advertisements are sent periodically and on request, by routers on all their configured interfaces. A router advertisement is sent to the all-nodes multicast address. This information that might be contained in the message:

- One or more prefixes that can be used on the link. This information enables stateless autoconfiguration of the hosts. These prefixes must be /64 for stateless autoconfiguration.
- Lifetime of the prefixes. By default, in Cisco IOS Software, the lifetime is very long: The default valid lifetime is 30 days, and the default preferred lifetime is 7 days.
- Flags that indicate the kind of autoconfiguration that the hosts can perform.
- Default router information, such as existence and lifetime.
- Other types of information for hosts, including default maximum transmission unit (MTU) and hop count.

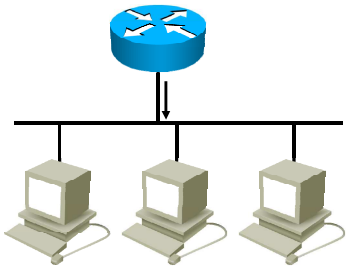
By sending prefixes, a router advertisement enables the autoconfiguration of hosts. By assigning lifetimes to prefixes, a router advertisement enables the renumbering of hosts. An old, deprecated prefix has a lifetime that is decreased to zero, and a new prefix will have a normal lifetime.

Router advertisement timing and other parameters can be configured on the routers.

Router Advertisement Parameters

Router advertisements:

- Default router
- IPv6 network prefix
- Options:
  - Lifetime of advertisement
  - MTU
  - Prefix Length
  - Router priority
  - L-bit
  - A-bit



Autoconfiguring IPv6 Hosts

The router plays an important role in host configuration of an IPv6 network. An IPv6 router uses the neighbor discovery protocol to periodically advertise information (router advertisements) to the all-nodes multicast address (FF02::1).

The router advertisements contain parameters that can be adjusted on the router. These parameters are specific to the interface on which they are configured, and consist of the following:

- The time interval between neighbor solicitations that the IPv6 nodes and the router use
- The time interval between periodic router advertisements
- The lifetime of the router advertisement
- The lifetime of the neighbor reachability cache
- The network prefix advertisement that the host autoconfiguration uses

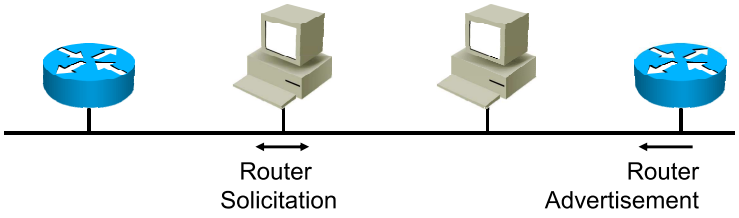
The router advertisement lifetime can be used to influence default router selection on a subnet. Ideally, a subnet that is served by multiple routers uses the best router as the default when that router is working; the default changes to another, less-optimal router if the best router stops working. You must carefully consider all candidate routers on the subnet when choosing the router lifetimes.

Additionally, the router sets these parameters:

- MTU
- Prefix length
- Router priority, which makes it possible to configure multiple routers with different priorities
- The L-bit flag that indicates whether the prefix is available on-link. On-link addresses can be accessed directly, not through a gateway. Note that lack of this flag does not mean that all addresses of the prefix are off-link.
- The A-bit flag that indicates whether the prefix may be used for autoconfiguration.

Router Advertisements

At boot time, nodes send router solicitations to promptly receive router advertisements.



Router solicitation packet:

- ICMP type 133
- Source = :: (unspecified address)
- Destination = FF02::2 (all-routers multicast address)

A router advertisement is typically sent immediately following a router solicitation. Router solicitations are sent by hosts at boot time, to ask routers to send an immediate router advertisement on the local link so that the host can receive the autoconfiguration information without waiting for the next scheduled router advertisement.

The router solicitation message is defined as follows:

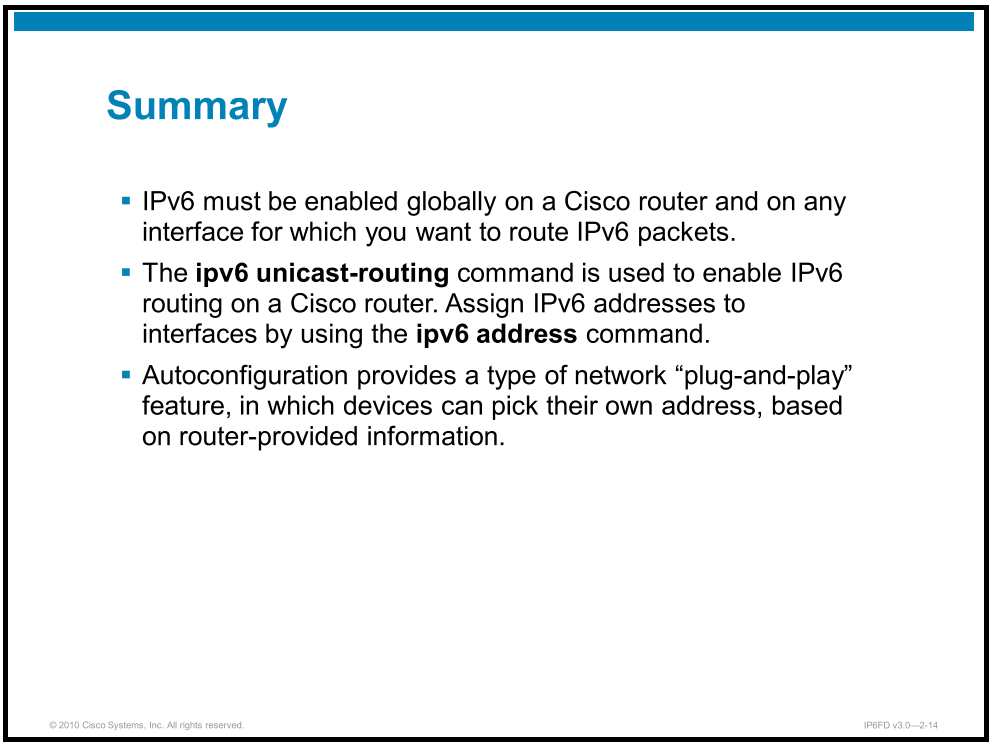
- The ICMP type is 133.
- The source address is the unspecified address (or the IP address that is assigned to the sending interface when the IP address is known, which is not usually the case).
- The destination address is the all-routers multicast address with the link-local scope.

When an answer to a router solicitation is sent, the destination address of the router advertisement is the unicast address of the requestor.

To avoid flooding, router solicitation should be sent only at boot time and only three times. This practice avoids flooding of router solicitation packets in the absence of a router on the network.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 5

Using ICMPv6 and Neighbor Discovery

Overview

Internet Control Message Protocol (ICMP) plays an important role in troubleshooting networks, facilitating simple tools such as **ping** or determining that a packet could not reach its destination. ICMP messages have also been leveraged for abusive purposes, such as Denial of Service (DoS) attacks. As a result and as a preventative measure, many organizations block ICMP traffic at the edge of their network. In IP version 6 (IPv6), the use of ICMP messages continues, with an expanded role. Adopters of IPv6 need to understand how ICMP version 6 (ICMPv6) messages are used so that they can develop security policies that protect their networks while permitting maximum functionality.

Any device that attaches to a network goes through numerous processes to identify itself and to obtain services from the network. This premise is true in either an IP version 4 (IPv4) or IPv6 network. However, people who design and manage IPv6 networks will discover that although the processes that are used in IPv6, have some similarities to those that are used in IPv4, they are different. Understanding these processes is fundamental to properly supporting an IPv6-enabled environment.

This lesson describes ICMP types and codes and IPv6 neighbor discovery, which is the process in which neighbors discover each other and autoconfigure addresses.

Objectives

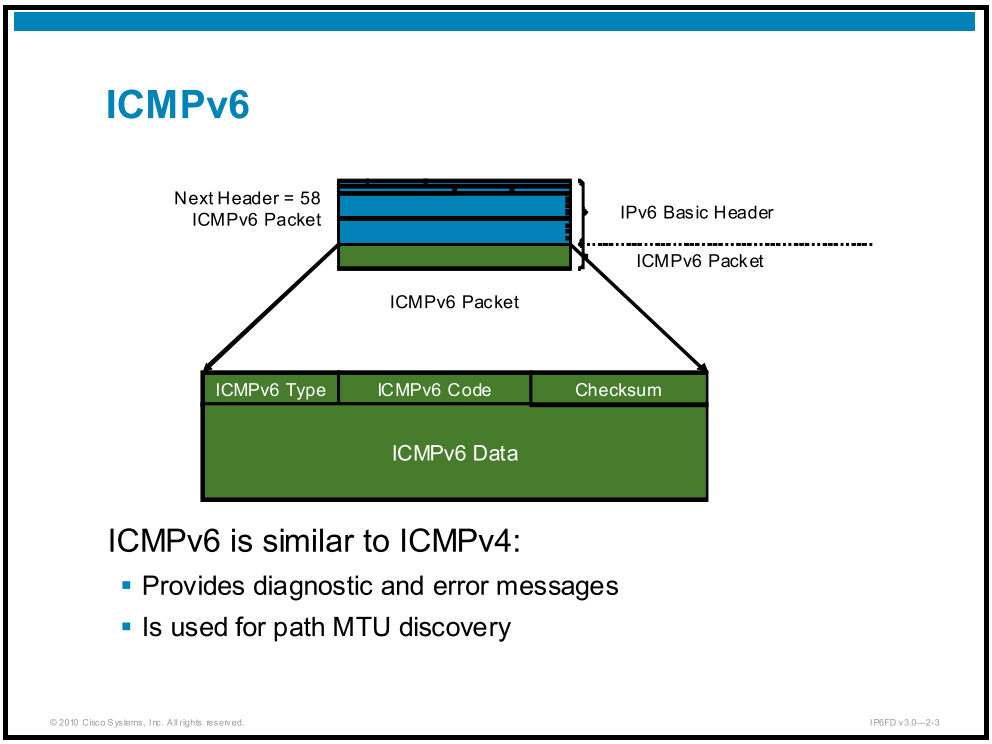
Upon completing this lesson, you will be able to describe ICMPv6 message types and how they are used to troubleshoot IPv6 issues, and you will be able to describe the neighbor discovery protocol. This ability includes being able to meet these objectives:

- Describe the format and use of ICMPv6 packets
- Describe the ICMPv6 error types and their codes
- Describe the ICMPv6 Echo Request and Echo Reply types
- Describe the data link layers for which IPv6 is defined
- Describe ICMPv6 neighbor discovery message types
- Describe how IPv6 stateless autoconfiguration works
- Discuss the value of autoconfiguration in IPv6

- Describe how renumbering is accomplished through router advertisements in IPv6
- Describe the syntax of the Cisco IOS commands that are used for neighbor discovery
- Describe a network prefix renumbering scenario in Cisco IOS Software
- Describes the ICMPv6 MLD message types
- Describe ICMPv6 message types that are used for IPv6 Mobility

ICMPv6

This topic describes the format of the ICMPv6 packet.



ICMPv6 is like ICMP version 4 (ICMPv4). ICMPv6 enables nodes to make diagnostic tests and report problems. Like ICMPv4, ICMPv6 implements two kinds of messages: error messages, such as Destination Unreachable, Packet Too Big, or Time Exceeded, and informational messages, such as Echo Request and Echo Reply.

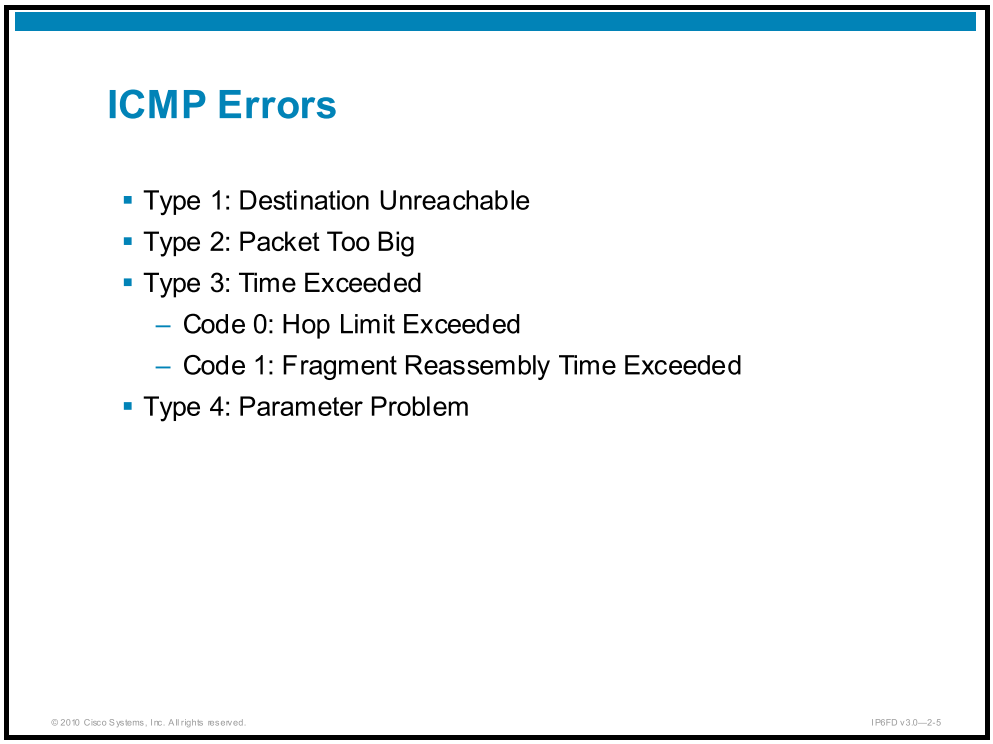
The ICMPv6 packet is identified as 58 in the Next Header field. An ICMPv6 packet is like a transport layer packet in the sense that it is at the end of the chain of extension headers, and it is the last chunk of information in the IPv6 packet. However, ICMPv6 is part of IPv6: ICMPv6 is not a Layer 4 protocol. Inside the ICMPv6 packet, the Type field identifies the type of ICMP message. The Code field further details the specifics of this type of message. For the receiver to check the integrity of the ICMPv6 packet, the Checksum field is computed over the ICMPv6 packet as well as some fields in the IPv6 header. The Data field contains information that is sent to the receiver for diagnostics or information purposes.

ICMPv6 is used in the path maximum transmission unit (MTU) mechanism, in which an ICMPv6 message of type Packet Too Big is sent back to the path MTU discoverer. This message contains the MTU of the next link. This process enables the path MTU mechanism at the origin to resend the packet with the received MTU from the ICMPv6 message.

ICMPv4 is often blocked by security policies in corporate firewalls because of known attacks that are based on ICMP. ICMPv6 is no different in this context, but it has the ability to use IP Security (IPsec) authentication and encryption if a security association exists between the parties. These security services decrease the possibilities of an attack that is based on ICMPv6.

ICMP Errors

This topic describes the ICMPv6 error types and their codes.



An ICMP type 1 error message is used to report destination-unreachable conditions. The code provides granularity to the source node, to determine why the packet could not reach its destination.

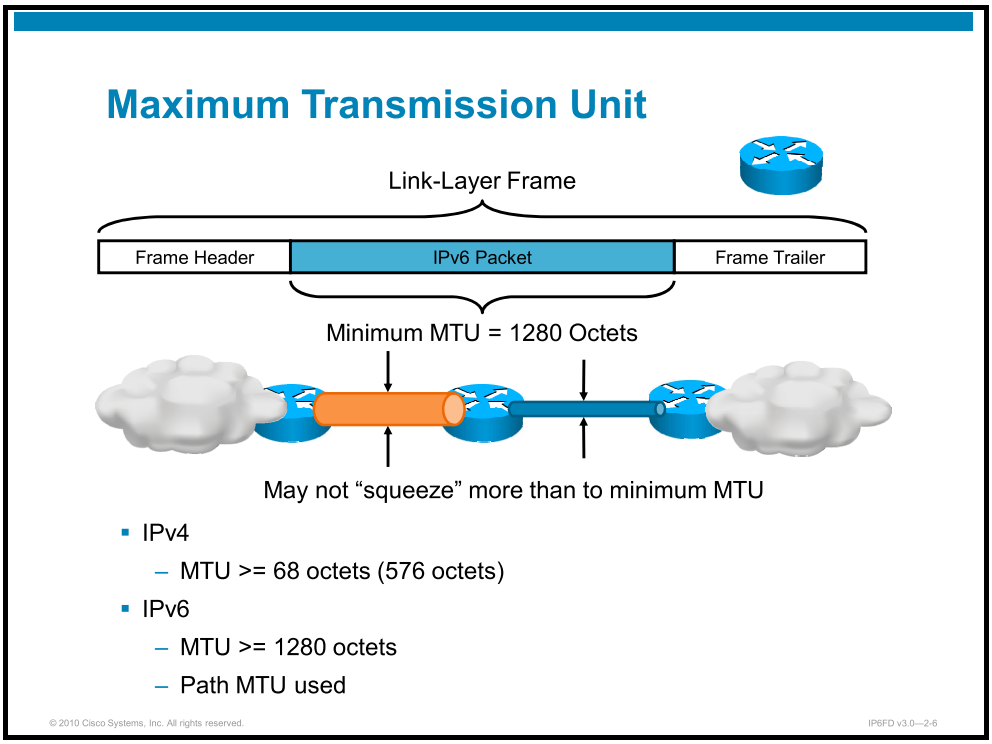
Tip	ICMPv6 error messages can be sent back to the source node by any intermediate node on the network. Unlike a successful Echo Request/Echo Reply exchange, which is end-to-end, ICMPv6 error messages are sent by the node that encounters the problem. Therefore, if a packet is undeliverable at any point in the path, that node will use its own IPv6 source address to send the error message, which has impacts on firewalls and other infrastructure devices.

An ICMP type 2 error message is an integral piece of the Path MTU Discovery (PMTUD) process and should not be blocked on a network.

Tip	IPv4 also uses ICMP messages to implement PMTUD. The difference is that in IPv4 the node that sends the Packet Too Big message cannot specify the optimal forward MTU of the next hop but can only report that the current packet is too large.

An ICMP type 3 error message indicates Time Exceeded, which in one case means that the hop limit has been reached. There are two codes: 0 indicates Hop Limit Exceeded, and 1 indicates Fragment Reassembly Time Exceeded.

An ICMP type 4 error message indicates Parameter Problem. As with other ICMP error messages, a portion of the invoking packet that caused this error is included in the ICMP message.

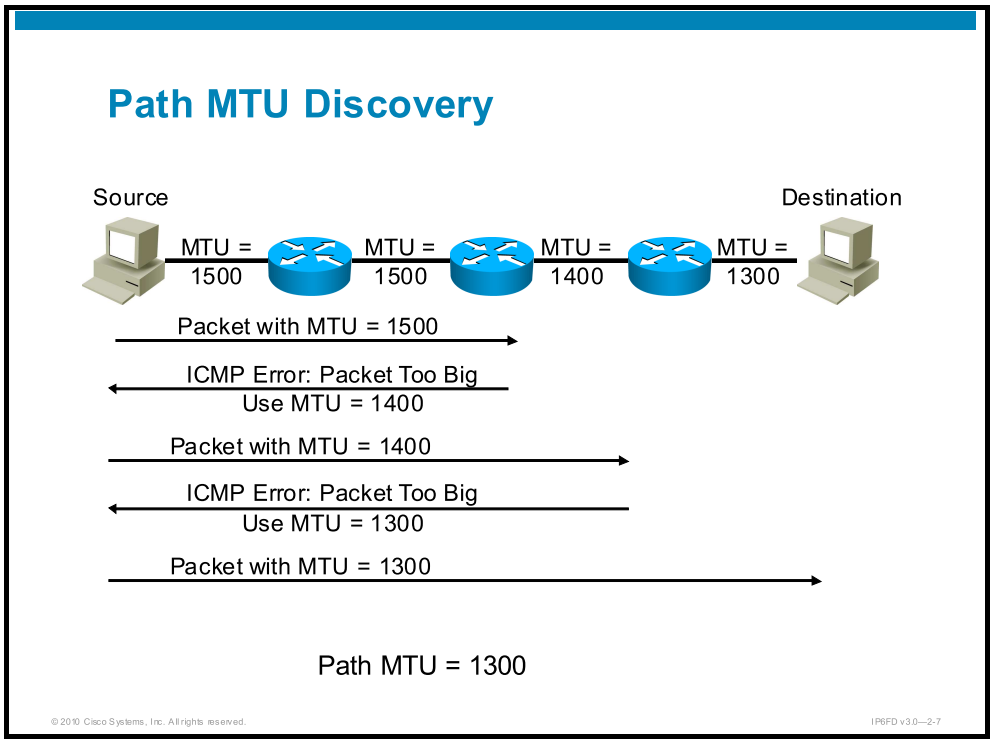


In IPv4, the specified minimum MTU is 68 octets; the recommended best-practice minimum is 576 octets, which is the minimum reassembly buffer size. Therefore, any IPv4 network must carry a packet that is as long as 68 octets. In IPv6, the minimum MTU is 1280 and the recommended minimum MTU is 1500, as a minimum reassembly buffer size.

Using PMTUD to find the maximum MTU in a path between the source and the destination is strongly recommended.

The basic IPv6 header supports a maximum packet size of 64,000 octets. However, larger packets are possible through a hop-by-hop option called the jumbogram.

IPv6 requires every link in the network to have an MTU of 1280 octets or greater. On a link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided to make the limitation transparent to IPv6.



PMTUD is a process to find the MTU that can be used from the source through the destination. The process is as follows:

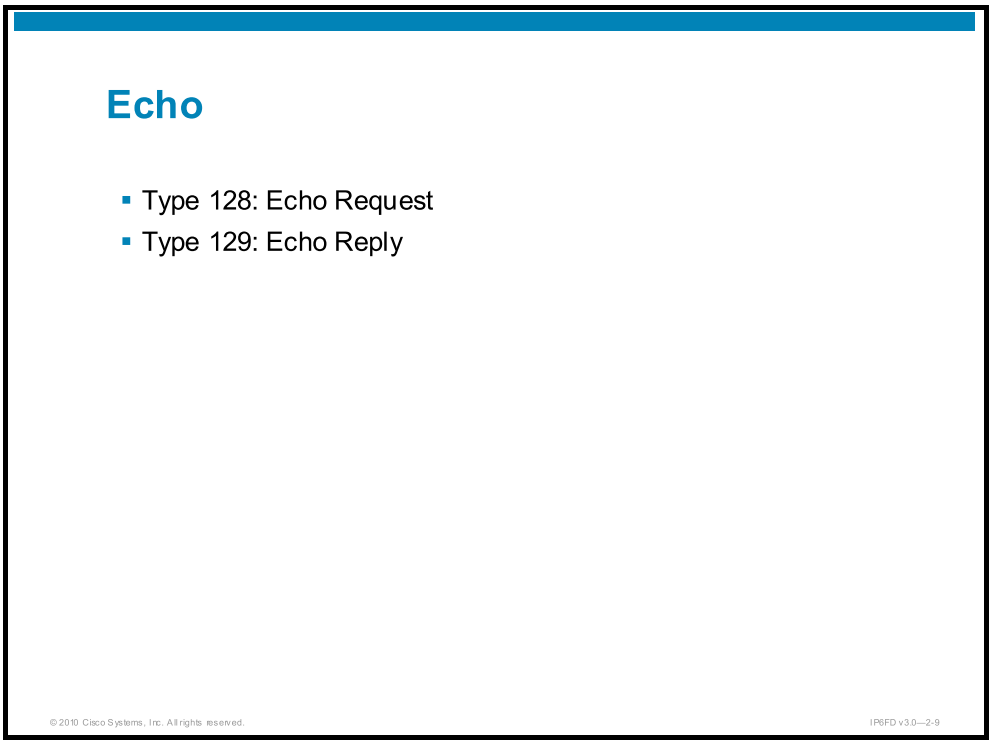
1. The source node sends a packet of a size equal to the MTU of its data link layer. In this example, 1500 is used.
2. The packet is forwarded through the network up to the destination, unless there is a smaller MTU in the path. If there is a smaller MTU in the path, the router sends back to the source node an ICMP type 2 error message (Packet Too Big). The content of the ICMP packet includes the MTU of the next data link layer, which is smaller than the size of the source packet. In this example, the next MTU is 1400, so the ICMP error message contains 1400 as a hint for the source.
3. The source node resends a packet that is equal to the size of the received MTU.
4. This process repeats until the packet reaches the destination. The size that is used in this packet is the path MTU. In this example, the path MTU is 1300.

IPv6 does not implement fragmentation in routers. Fragmentation is done only by the source node when the path MTU is not large enough. In IPv4, PMTUD is optional and seldom used. (IPv4 implements PMTUD by using the Don't Fragment, or DF, bit.)

Tip Nodes and applications do not need to make use of PMTUD but rather can simply send the Guaranteed Not To Be Too Big packet of 1280 bytes. For certain applications, finding the largest packet size possible between two peers (moving large data stores, for example) makes sense. For others, when sending small packets might even be preferable (line VoIP), finding the largest packet size possible might not be worth the discovery time and overhead.

Echo

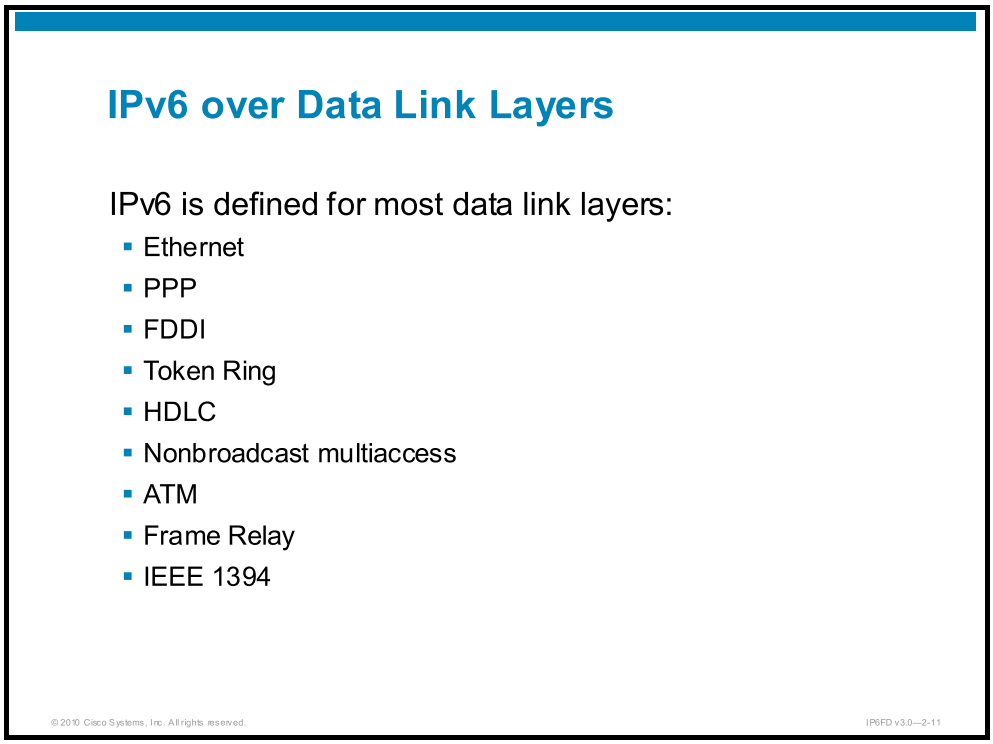
This topic describes the ICMPv6 Echo Request and Echo Reply types.



The ping application uses Echo Request packets to probe for active systems. The Echo Reply packet is the return response that is sent by a node that receives the Echo Request.

IPv6 over Data Link Layers

This topic describes the data link layers for which IPv6 is defined.

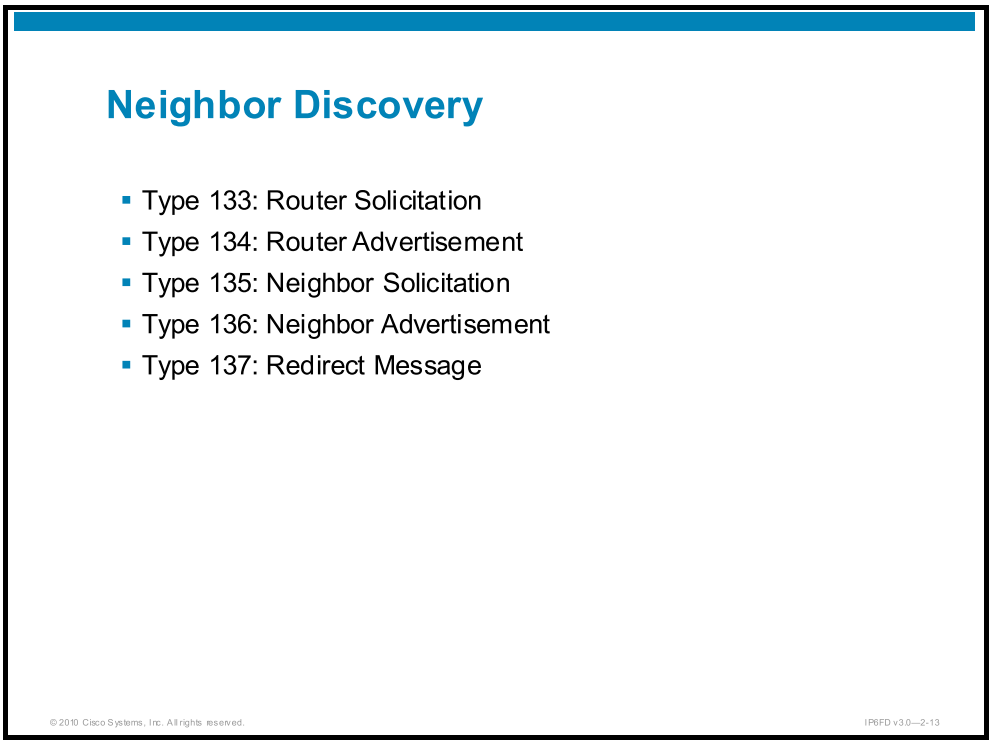


IPv6 is defined on most of the current data link layers. Therefore, an RFC describes the behavior of IPv6 in each of these specific data link layers. Cisco IOS Software does not necessarily support all of those options.

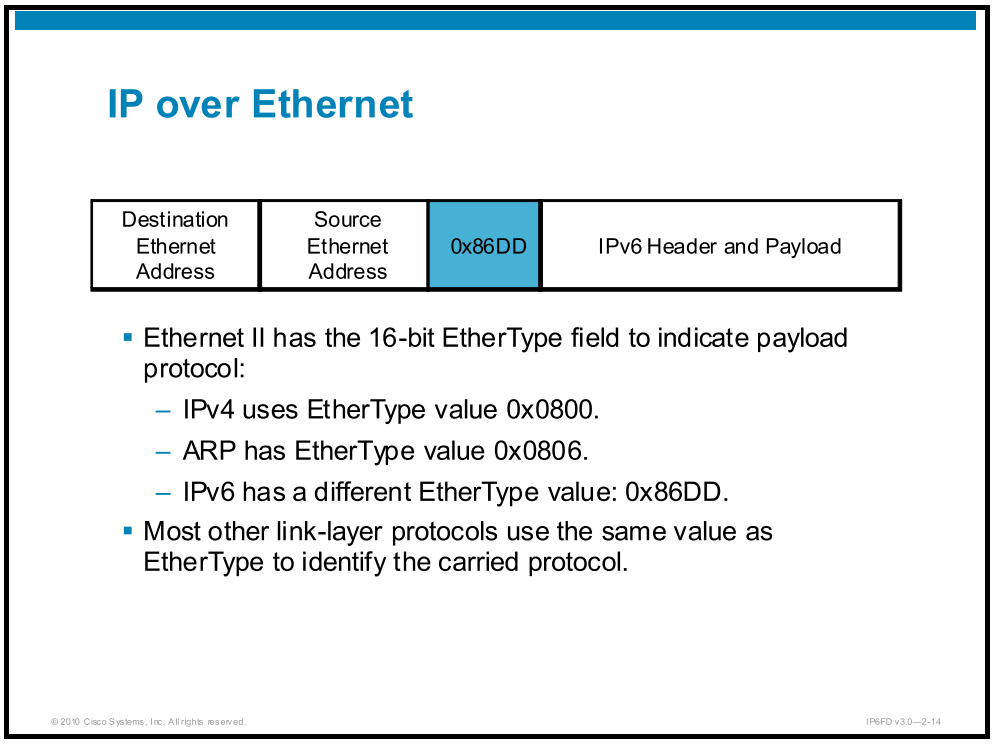
The data link layer defines how IPv6 interface identifiers are created.

Neighbor Discovery

This topic describes ICMPv6 neighbor discovery message types.



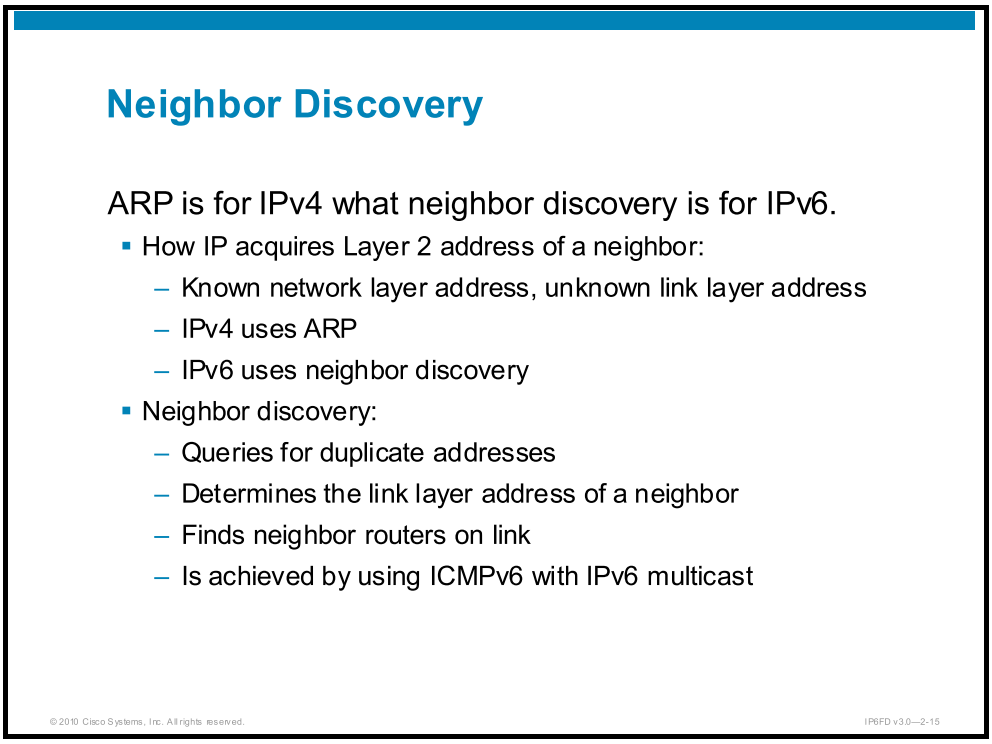
Neighbor discovery is used on-link for router solicitation and advertisement, for neighbor solicitation and advertisement (acquisition of data link layer addresses for IPv6 neighbors), and for the redirection of nodes to the best gateway.



Ethernet II uses the EtherType field to identify the network layer protocol.

IPv6, like any network layer protocol, can be used on Ethernet. IPv6 has its own Ethernet protocol ID, 0x86DD, which differentiates the packet from other protocols. Inside the Ethernet frame are the IPv6 headers and payload.

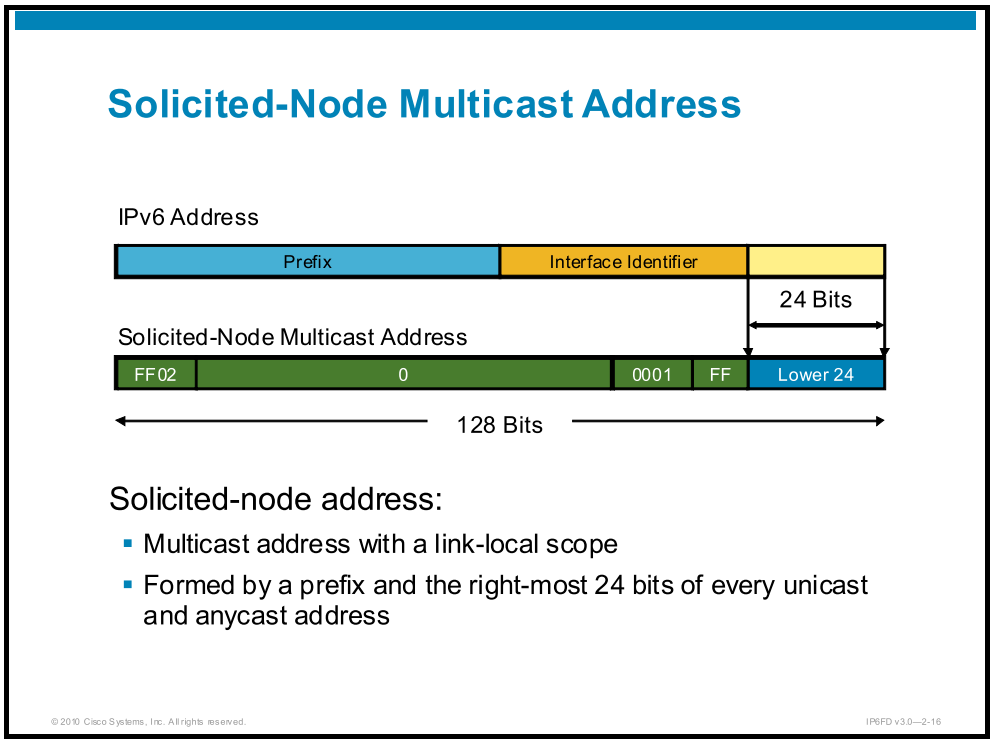
IPv6 interface identifiers for Ethernet use the extended universal identifier (EUI)-64 IEEE standard.



Neighbor discovery is a process that enables these functions:

- Determine the data link layer address of a neighbor on the same link, like the Address Resolution Protocol (ARP) does in IPv4.
- Find neighbor routers.
- Keep track of neighbors.

Neighbor discovery achieves these results by using ICMP with multicast addresses.

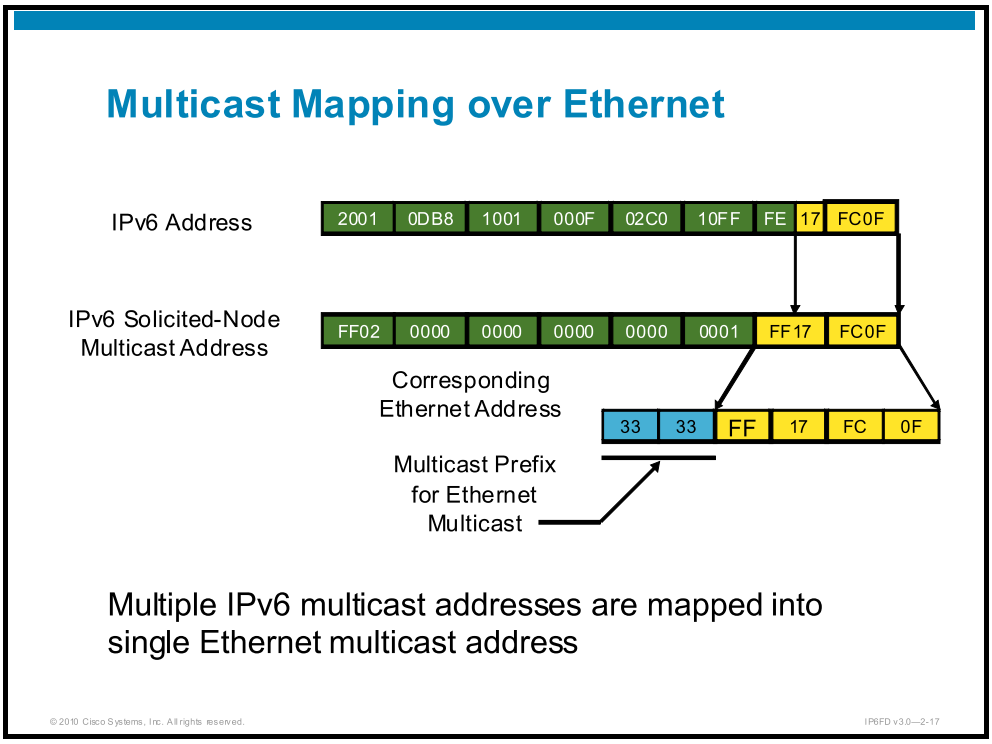


The solicited-node address is a multicast address. Any node must join the multicast group that corresponds to each of its unicast and anycast addresses. The solicited-node address is composed of the FF02:0:0:0:0:1:FF/104 prefix, concatenated with the right-most 24 bits of the corresponding unicast or anycast address.

As an FF02::/16 address, a solicited-node multicast address has a link-local scope.

Solicited-node addresses are used for Neighbor Solicitation messages, when another node needs the data link layer address of an IPv6 address to send the right frame on the data link layer. The source node takes the right-most 24 bits of the IPv6 address of the destination node and sends a Neighbor Solicitation message to the multicast group on the link-local address. The corresponding node responds with its data link layer address.

This function avoids the broadcasts that are used in IPv4 ARP, in which all nodes receive the requests.

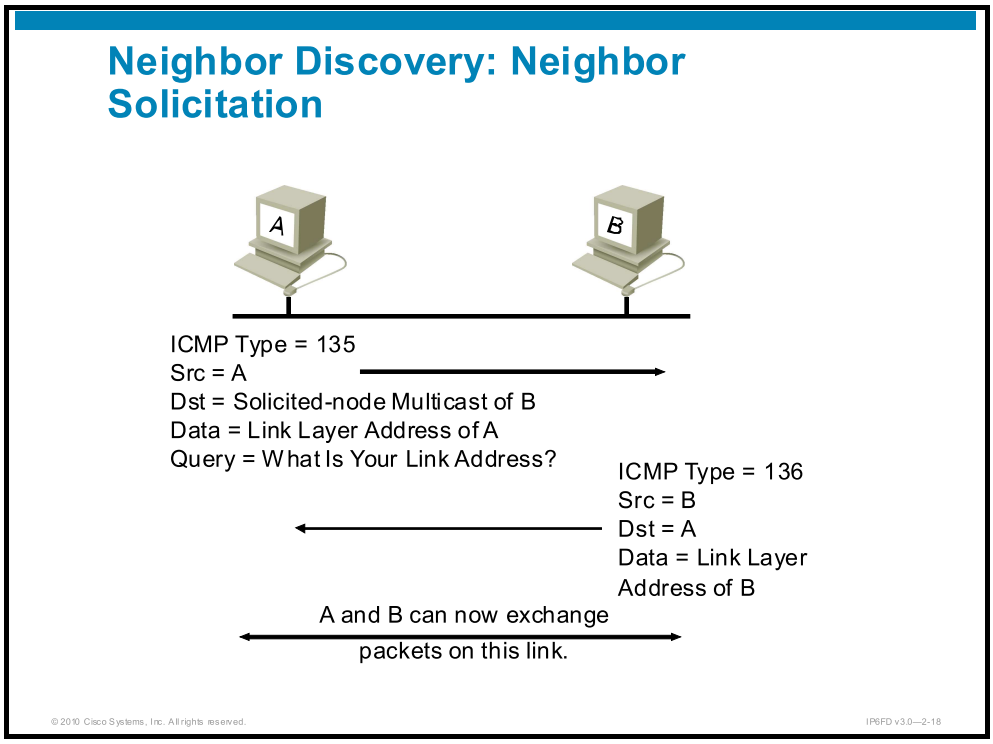


If an IPv6 address is known, then the associated IPv6 solicited-node multicast address is known. The example in the figure gives the IPv6 address 2001:DB8:1001:F:2C0:10FF:FE17:FC0F. The associated solicited-node address is FF02::1:FF17:FC0F.

If an IPv6 multicast address is known, then the associated Ethernet MAC address is known. Multicast Ethernet addresses are formed by adding the last 32 bits of the IPv6 multicast address to 33:33.

As the figure shows, the IPv6 solicited-node multicast address is FF02::1:FF17:FC0F. The associated Ethernet MAC address is 33:33:FF:17:FC:0F.

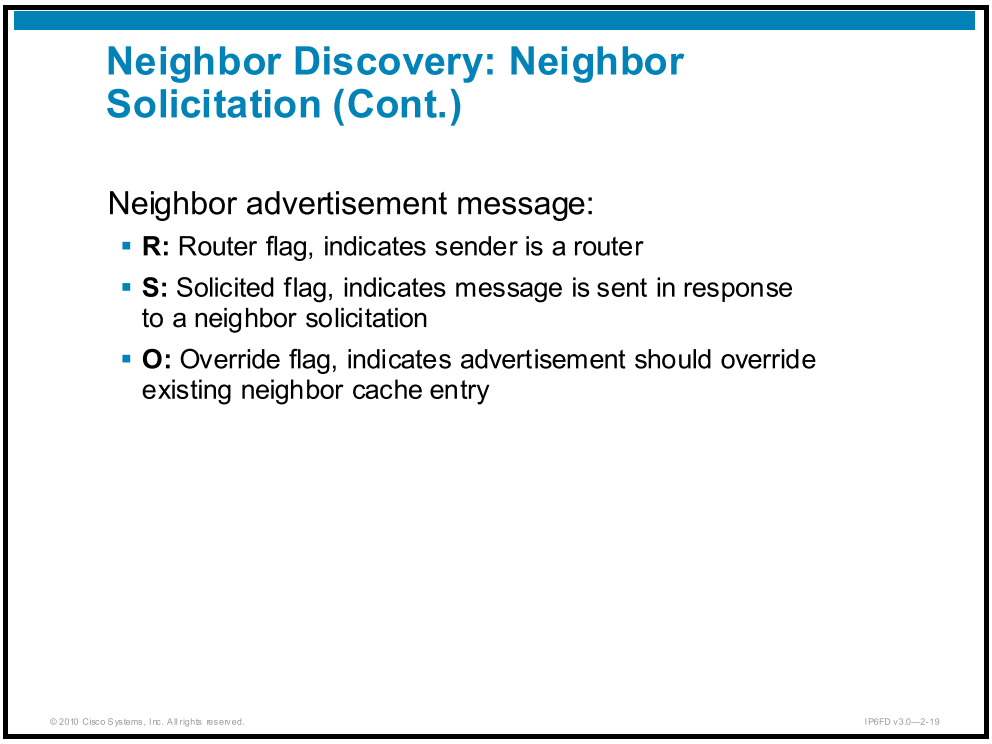
Tip You must understand that the resulting MAC address is a virtual MAC address: It is not burned into any Ethernet card. Depending on the IPv6 unicast address, which determines the IPv6 solicited-node multicast address, any Ethernet card may be instructed to listen to any of the 2^{24} possible virtual MAC addresses that begin with 33-33-FF. In IPv6, Ethernet cards often listen to multiple virtual multicast MAC addresses as well as their own burned-in, unicast MAC addresses.



The example in the figure shows how to determine the data link layer address of a neighbor in a process called Neighbor Solicitation. This function is like ARP in IPv4. For efficiency, the requestor also sends the data link layer address of the source node.

The ICMP message type 135, which is identified as Neighbor Solicitation, is sent on the link. The source address is the IPv6 address of the source node, if known. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The message also includes the data link layer address of the source node, so that the destination node can use that address right away.

The destination node responds with an ICMP message type 136, which is identified as Neighbor Advertisement, on the link. The source address of the responding message is the IPv6 address of the destination node, and the destination address is the IPv6 address of the source node, because it is the answer. The data portion includes the data link layer address of the destination node, which is redundant, and the data link layer address that is included in the frame. After receiving the answer, the source node and the destination node can communicate on the link because the data link layer addresses are known to both.

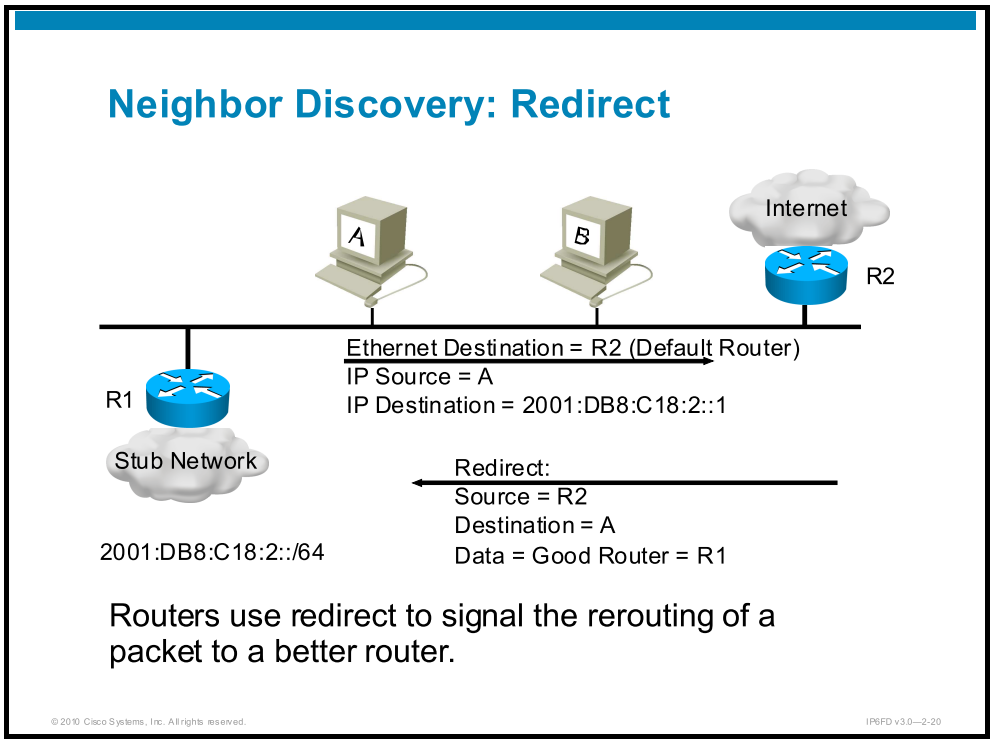


The neighbor solicitation process is also used to verify the reachability of a neighbor. In that case, the destination address is not the multicast address but the unicast address of the neighbor. Neighbor advertisements can be sent when there are changes in the data link layer addresses. In that case, the source of the advertisement sends a message to the all-node multicast address.

Neighbor advertisements contain three flags to indicate the purpose of the message:

- **R flag:** Indicates that the sender is a router
- **S flag:** Indicates that the message was sent as a response to a previous neighbor solicitation message
- **O flag:** Indicates that the information in the advertisement should be used to override the existing entry in the neighbor cache

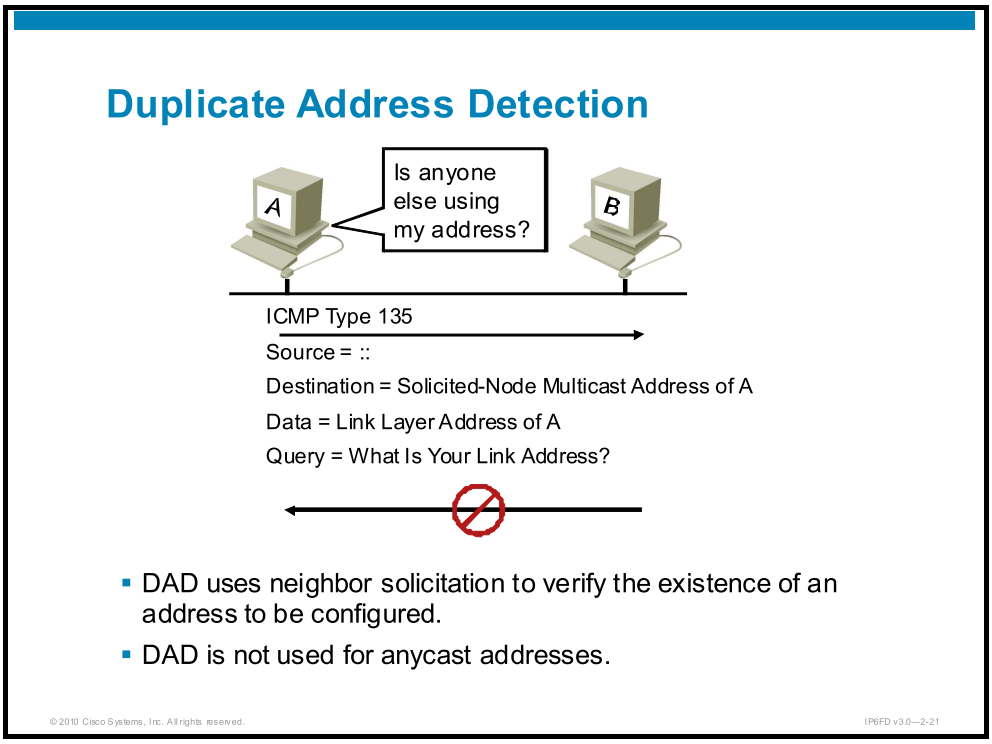
The last two flags are used to craft a response with the same purpose as gratuitous ARP in IPv4.



A Redirect Message is sent by a router to signal the rerouting of a packet to a better on-link router. The host that receives this message then reroutes future packets via the better router. This process is equivalent to the ICMP redirect function in IPv4.

In this example, node A sends a normal IPv6 packet to a destination address, 2001:DB8:C18:2::1, by its default router, R2. R2 forwards the packet to R1 as normal but also knows that R1 has an interface on the same segment as node A. R2 then also sends a Redirect Message back to node A, directing the use of R1 to reach the targeted destination for future packets.

Tip Router Redirect Messages can pose a security threat to hosts that share a link with the attacker. An attacker can send a manufactured Redirect Message, as if it came from the router, to the victim, telling it to use a different first-hop address to reach off-link destinations. The traffic of the victim is often redirected to the attacker. At that point, the attacker can either black-hole the traffic (DoS attack) or forward the traffic, resulting in a man-in-the-middle attack.



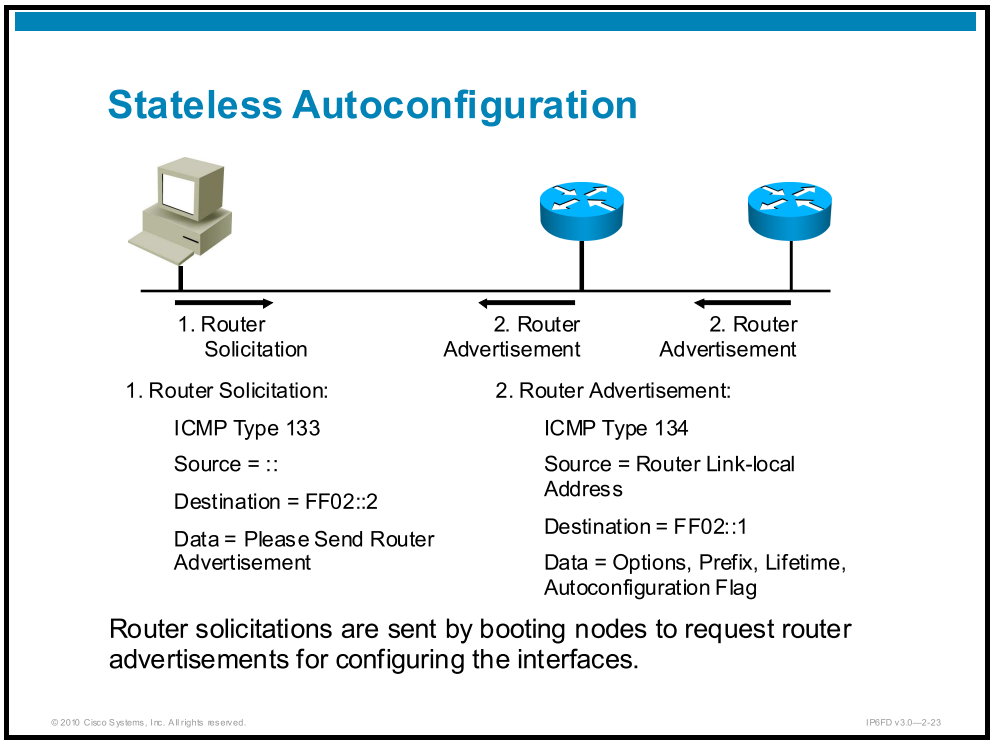
Duplicate Address Detection (DAD) uses neighbor solicitation to query whether another node on the link has the same IPv6 address. DAD sends a Neighbor Solicitation packet to the solicited-node multicast address of its own IPv6 address. The source address of this packet is the unspecified address (::). If a node responds to that request, then the IPv6 address is in use and the requesting node should not use that address.

DAD is used during the autoconfiguration process to make sure that no one else is using the autoconfigured address.

Tip Note that the DAD test is sent from the unspecified address. This is the case when the node has yet to initialize a link-local address and has no valid addresses yet—even the link-local address is marked as tentative. If the link-local address is already configured for an interface, and the interface needs to run the DAD process on another address, the neighbor solicitation is sent from the already valid and tested link-local address (or another valid address) because DAD is a strictly link-local process.

Stateless Autoconfiguration

This topic describes how IPv6 stateless autoconfiguration works.



Stateless autoconfiguration is an important feature of IPv6. This type of autoconfiguration allows serverless, basic configuration of the nodes, as well as easy renumbering.

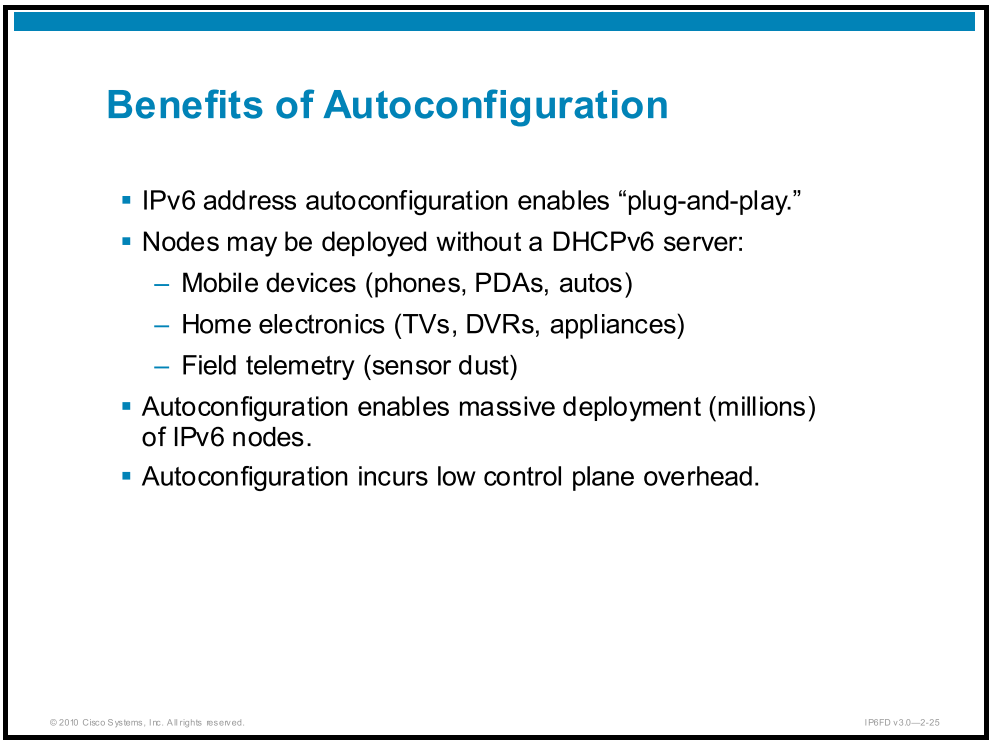
Stateless autoconfiguration uses the information in the Router Advertisement messages to configure the node. The prefix that is included in the router advertisement is used as the /64 prefix for the node address. The other 64 bits are obtained by the dynamically created interface identifier, which in the case of Ethernet is the modified EUI-64 format.

Router advertisements are sent periodically. When a node boots, the node needs its address in the early stage of the boot process. Instead of waiting for the next router advertisement to get the information to configure its interfaces, a node sends a Router Solicitation message, asking the routers on the network to reply immediately with a router advertisement so that the node can immediately autoconfigure. All the routers respond with a normal Router Advertisement message that has the all-nodes multicast address as the destination address.

Stateless autoconfiguration, as it exists today, does not address the issue of finding the Domain Name System (DNS) server for DNS resolution or of registering the computer in the DNS space. The Internet Engineering Task Force (IETF) is discussing these issues to enhance the autoconfiguration function. A draft that would allow DNS information to be specified in router advertisements is under discussion.

Value of Autoconfiguration

This topic discusses the value of autoconfiguration in IPv6.

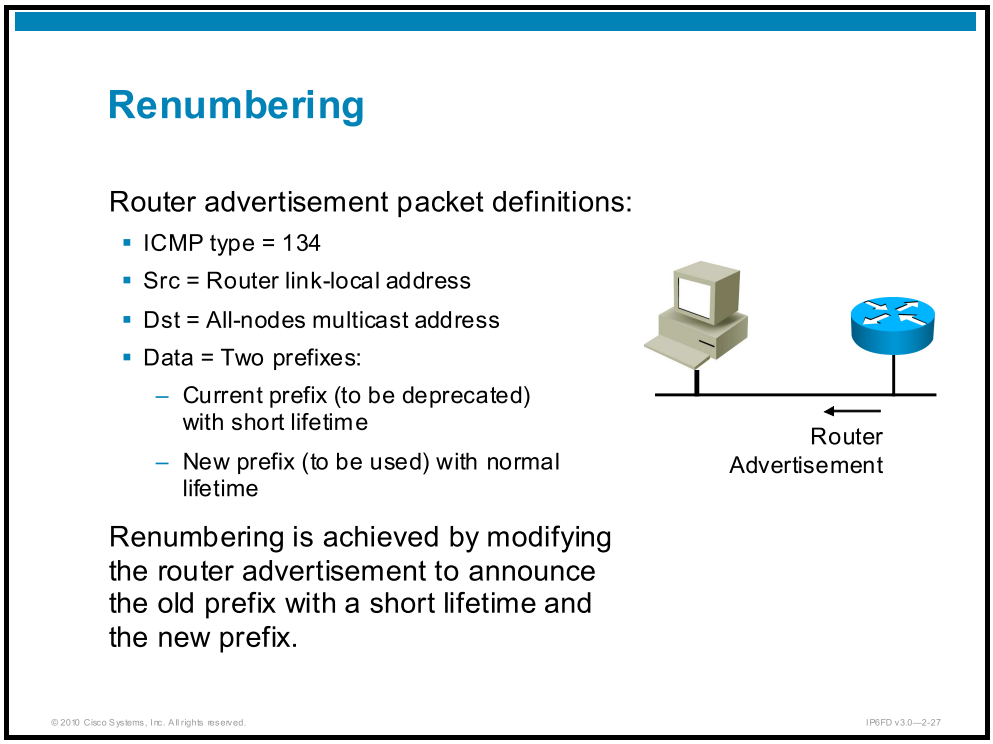


Autoconfiguration in IPv6 simplifies the deployment of nodes that might not have access to a DHCP version 6 (DHCPv6) server. Also, devices that use low-memory-capacity integrated circuit chips, commonly referred to as thin clients, might not have adequate space to accommodate the programming code that supports complex connection processes. Autoconfiguration is a basic function of any IPv6-enabled network stack. Autoconfiguration facilitates the deployment of new classes of IPv6-enabled devices, such as low-power field sensors for the military, new mobile devices, and home appliances.

More-traditional network environments, specifically enterprise-type networks, will probably continue to rely on well-known mechanisms—namely DHCP—for IP address allocation. However, in environments that have a high volume of transient nodes, such as airport hotspots or cellular networks, autoconfiguration reduces the overall complexity of servicing millions of nodes.

Renumbering

This topic describes how renumbering is accomplished through router advertisements in IPv6.



Renumbering of nodes is achieved by sending router advertisements. These messages contain both the old prefix and the new prefix. Decreasing the lifetime of the old prefix tells the nodes to use the new prefix while keeping their current connections open to the old prefix. During that period, nodes have two unicast addresses to use. When the old prefix is no longer used, the router advertisement will include only the new prefix.

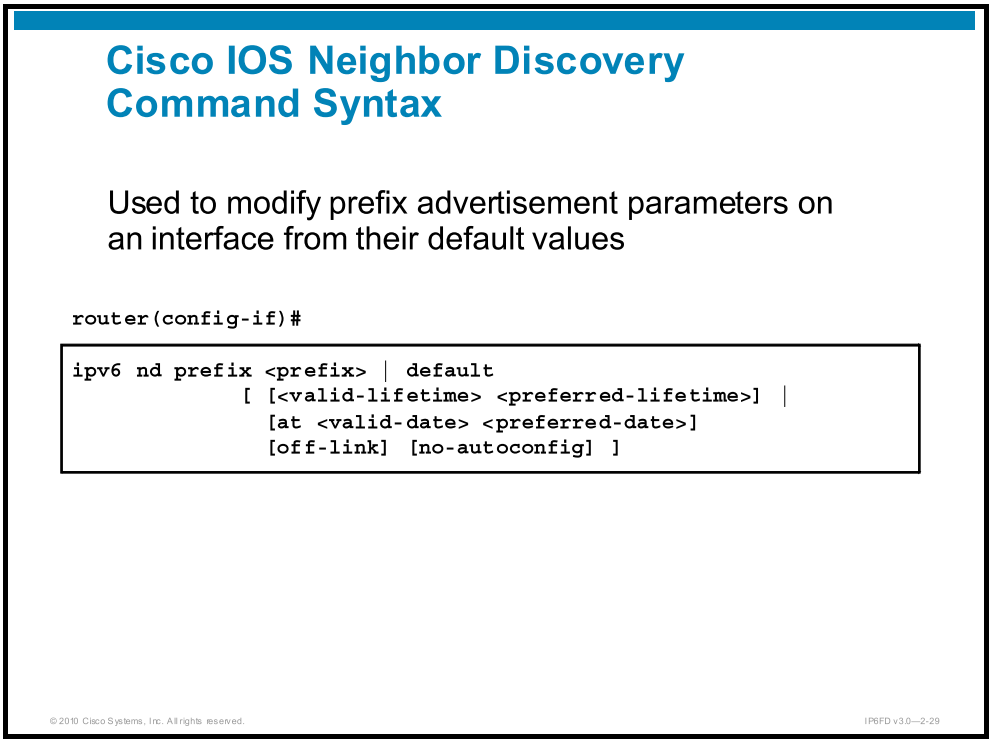
If stateless autoconfiguration is not used, then other ways of renumbering should be applied. Autoconfiguration greatly helps the renumbering process.

Renumbering of a whole site also means that the routers need to be renumbered. A router renumbering protocol is defined for that purpose in RFC 2894, *Router Renumbering for IPv6.*

Note Router renumbering, as described in RFC 2894, has not been widely implemented.

Cisco IOS Neighbor Discovery Command Syntax

This topic describes the syntax of the Cisco IOS commands that are used for neighbor discovery.



By default, all /64 prefixes that are configured as addresses on an interface are advertised in router advertisements. The **ipv6 nd prefix** interface command is used to explicitly specify the advertised prefixes in the router advertisement messages. The syntax of the command is as follows:

- **ipv6 nd prefix prefix <prefix> | default [[<valid-lifetime> <preferred-lifetime>] | [at <valid-date> <preferred-date>] [off-link] [no-autoconfig]]**

For example:

```
ipv6 nd prefix 2001:db8:c18:2::/64 43200 43200
```

The specified prefix is 2001:DB8:C18:2::/64. If the **default** keyword is used instead of the **prefix** keyword, then the specified parameters apply to all the prefixes.

In this example, the prefix is advertised on the link with 43,200 seconds as the valid and preferred lifetime. A host on the link that uses address autoconfiguration will autoconfigure its IPv6 address from the advertised prefix. The valid and preferred lifetime counters are set to the value that is specified in the router advertisement (43,200 seconds). Because the router sends periodic router advertisements, the valid and preferred lifetime counters on the host are periodically reset to the advertised values. Instead of specifying an explicit lifetime, a date can indicate the prefix expiration. The valid and preferred lifetimes are then counted down in real time, and when the expiration date is reached, the prefix is longer advertised.

Usually, an announced prefix is a valid on-link prefix. A node that sends traffic to such addresses considers the destination to be on the same link. If the off-link parameter is used for a given prefix, the announcement makes no statement about the on-link or off-link property of the prefix.

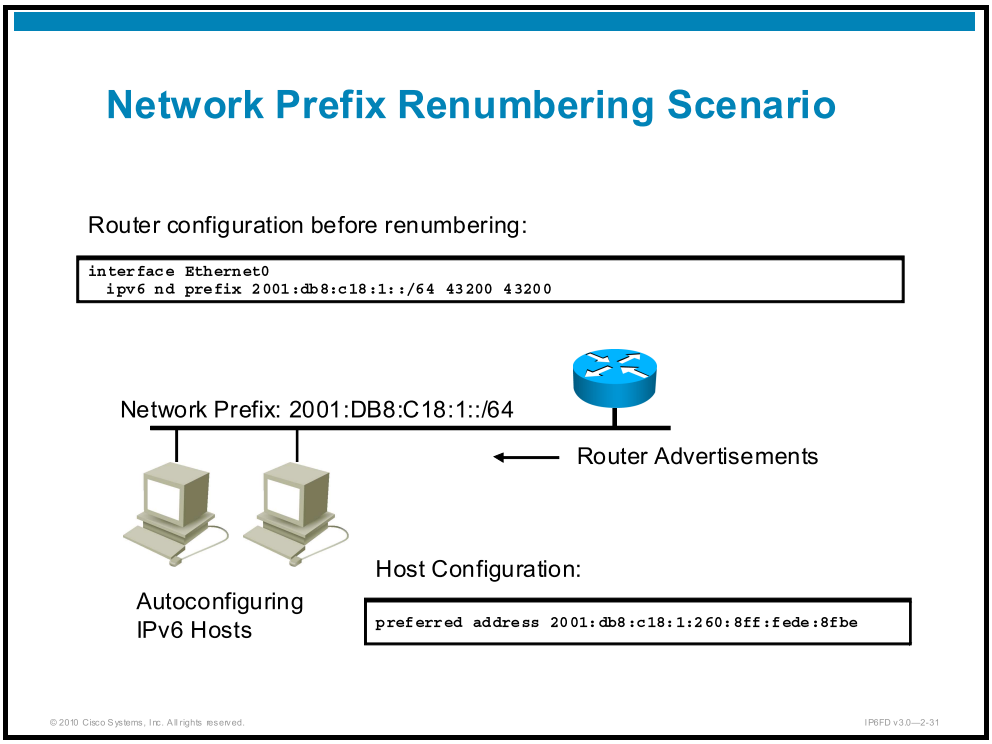
The **no-autoconfiguration** parameter indicates that the prefix cannot be used for address autoconfiguration.

By default, both on-link and autoconfiguration parameters are set.

The **ipv6 nd suppress-ra interface** command suppresses router advertisements from being sent. By default, Cisco routers send router advertisements on multi-access link types (such as Ethernet) and do not send router advertisements by default on point-to-point link types (such as serial links).

Cisco IOS Network Prefix Renumbering Scenario

This topic describes a network prefix renumbering scenario in Cisco IOS Software.



By default, the router advertises all prefixes (site-local, unique local, and global) that are configured on each interface. Many parameters—such as the router advertisement lifetime, time interval between router advertisements, and other neighbor discovery parameters—use a default value when nothing else is explicitly defined.

You can override the prefix advertisement default values so that only the specified prefix advertisements are sent in router advertisements. In the example that the figure shows, the default values of the prefix advertisement parameters are overridden by using the **ipv6 nd prefix** interface command. The two values that follow the prefix parameter are the valid lifetime and preferred lifetime addresses that are announced in the prefix. The valid lifetime is the length of time that an address remains in the valid state. After that time, the address becomes invalid.

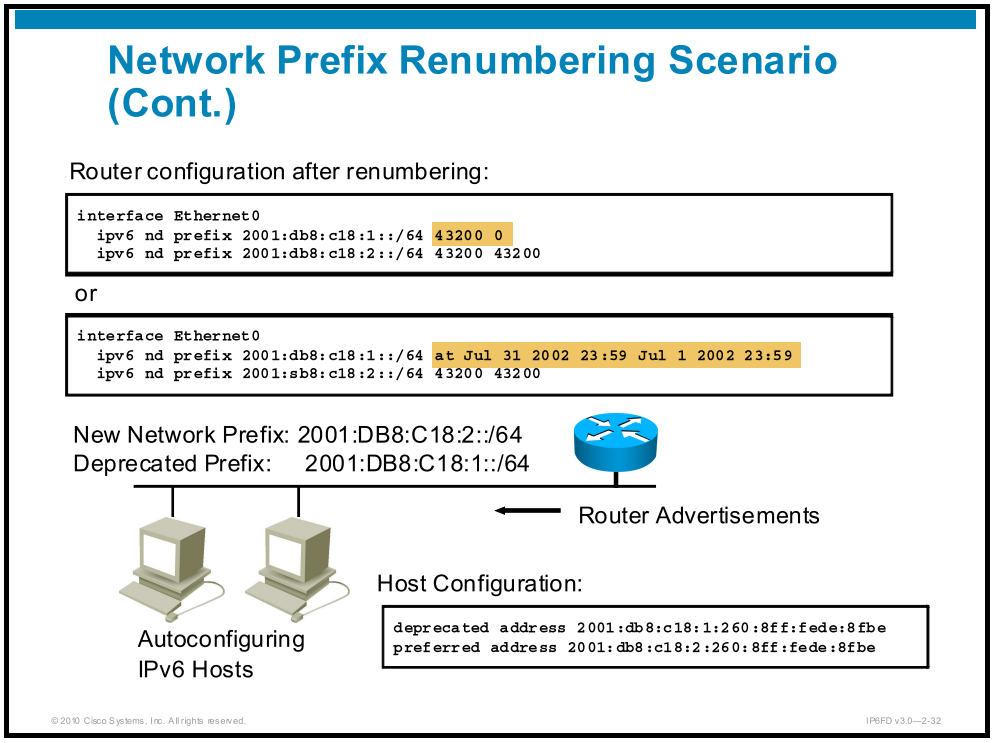
The preferred lifetime is the length of time that an address that is generated from the announced prefix via stateless address autoconfiguration remains preferred. The preferred lifetime must always be less than or equal to the valid lifetime.

Because router advertisements are sent periodically, hosts refresh the prefix information and lifetimes accordingly. The address that is generated from the advertised prefix becomes deprecated after the preferred lifetime expires. At that point, the host does not use the deprecated address for new network connections but might still accept connections to this deprecated address, as long as the valid lifetime has not expired. This property is used for site renumbering of IPv6 addresses.

The example shows a network in which the currently advertised network prefix is 2001:DB8:C18:1::/64. This prefix is advertised with a lifetime of 12 hours.

Suppose you need to renumber the hosts in the network to use a new prefix, such as 2001:DB8:C18:2::/64. To do so, you can leverage the benefits of router prefix advertisements.

Tip	This technique for renumbering is best called host renumbering. Normally, routers do not autoconfigure based on router advertisements. These and other network infrastructure components are usually configured through many static settings, including address assignment, route summarization and aggregation, access control list (ACL) or firewall rules, and other parameters. Stateless autoconfiguration helps when renumbering hosts, but enterprise renumbering is still a significant undertaking.



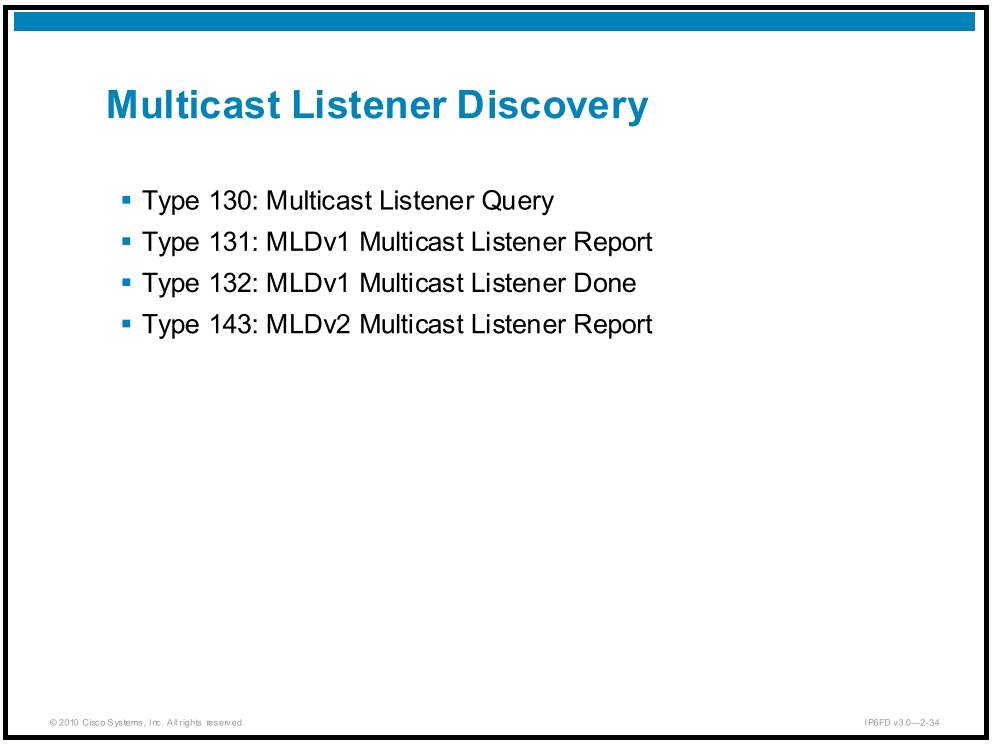
In this renumbering scenario, the network administrator decreases the preferred lifetime of the previous network prefix to zero, thus indicating to the hosts that they should deprecate the old network prefix. At the same time, a new network prefix is advertised.

The hosts in the network then have two configured network prefixes. The deprecated, or previous, network prefix is no longer be used to initiate new connections. The reason for deprecating the previous prefix instead of simply removing it (making the address invalid) is to allow existing connections to survive the renumbering process. If the advertised valid lifetime is set to zero, then hosts on the network cannot accept connections that are directed to the deprecated address.

You can eventually remove the deprecated address announcement.

ICMP MLD

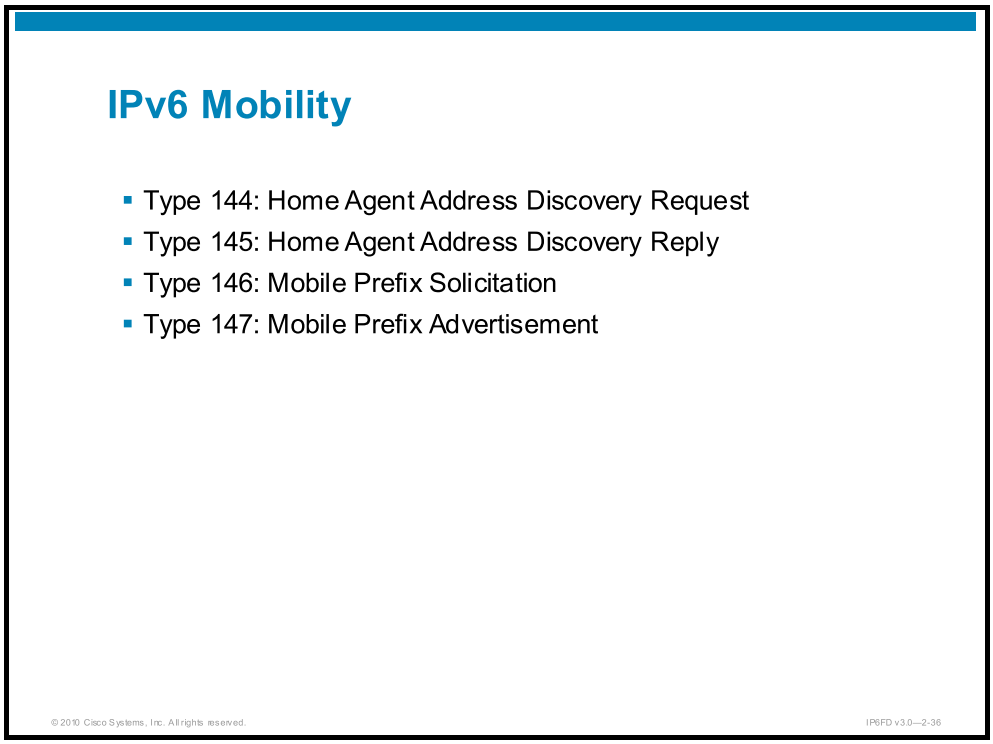
This topic describes the ICMPv6 Multicast Listener Discovery (MLD) message types.



MLD is used on-link, for routers to learn about multicast listeners. When a node joins a multicast group, it reports this join via an MLD version 1 (MLDv1) or version 2 (MLDv2) Report message.

IPv6 Mobility

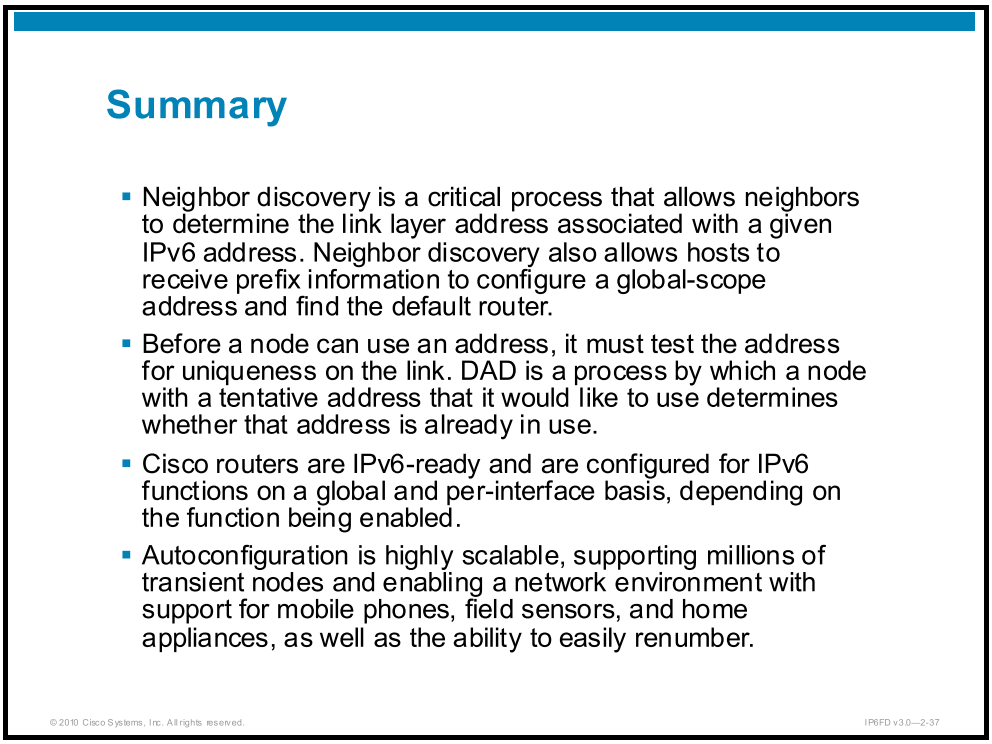
This topic describes ICMPv6 message types that are used for IPv6 Mobility.



Types 144 through 147 are used between the mobile node and home agent, to exchange mobility information.

Summary

This topic summarizes the key points that were discussed in this lesson.



Lesson 6

Troubleshooting IPv6

Overview

Each Cisco router that is used to support the integration of IP version 6 (IPv6) into a network needs to be configured by using a specific command set. When IPv6 has been enabled on a Cisco router, issues that require the troubleshooting of Cisco IOS Software configurations might arise. This lesson describes the IPv6 configuration process on Cisco IOS Software and provides some basic methods for troubleshooting issues that relate to IPv6 configurations.

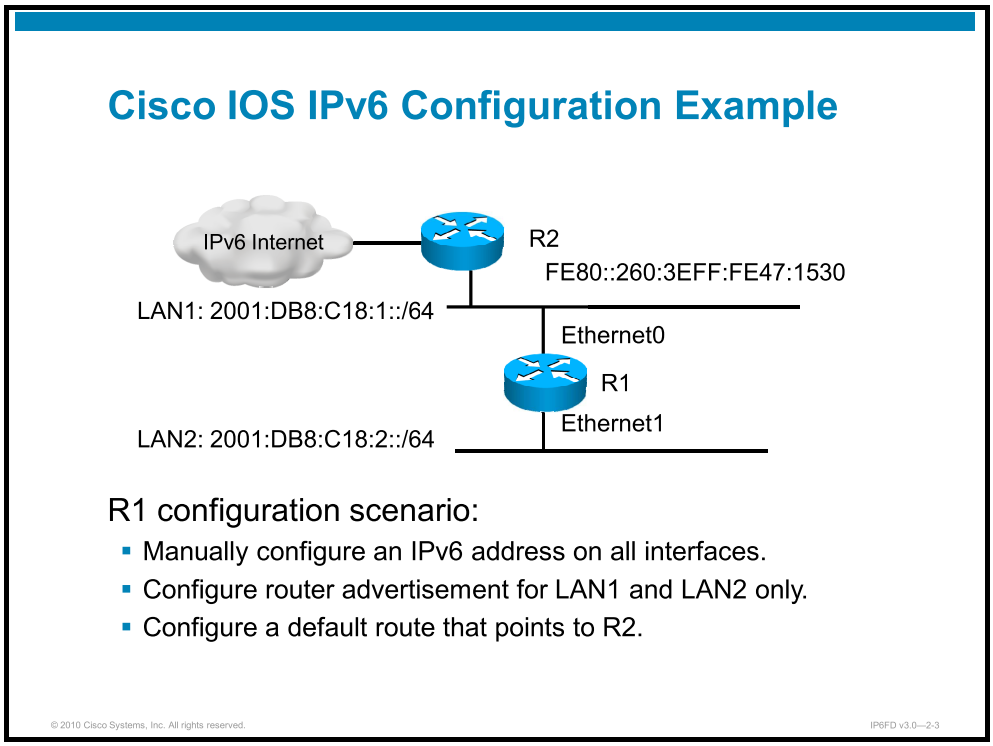
Objectives

Upon completing this lesson, you will be able to configure and troubleshoot a Cisco IOS router to support IPv6 operation. This ability includes being able to meet these objectives:

- Configure a Cisco router to support IPv6 operation
- Troubleshoot IPv6 configuration problems
- Describe some useful Cisco IOS Software IPv6 **debug** commands
- Describe sample output from the **debug ipv6 icmp** Cisco IOS Software command

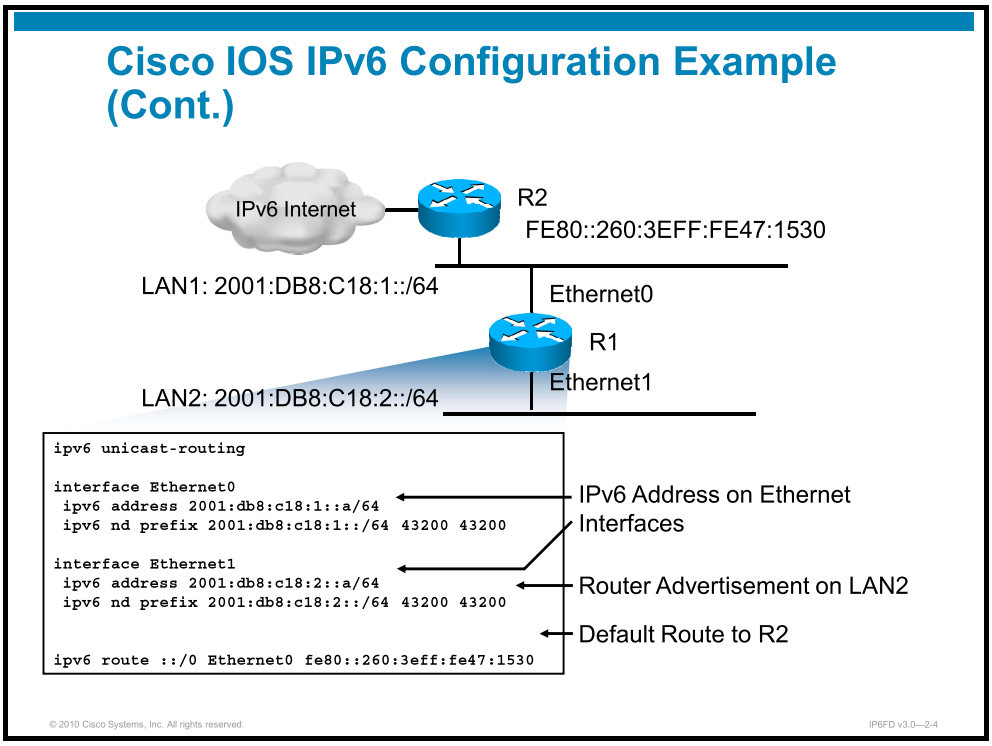
Cisco IOS IPv6 Configuration Example

This topic describes a Cisco IOS IPv6 configuration example.



The example includes an IPv6 configuration of router R1 in a basic network, as follows:

- R1 advertisements are activated on the Ethernet0 (LAN1) and Ethernet1 (LAN2) interfaces for the global IPv6 prefixes.
- R1 router announcements for both LANs use a prefix with valid and preferred lifetimes of 12 hours (43,200 seconds).
- An IPv6 default route that points to router R2 is installed on R1.



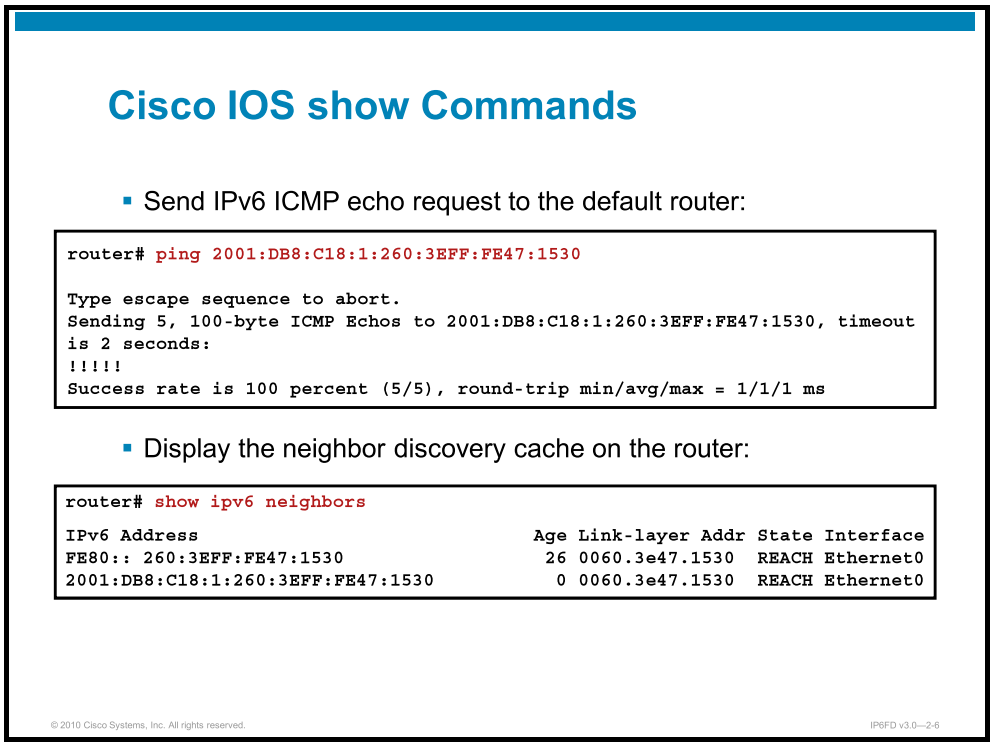
The figure shows the configuration of router R1. This configuration defines the IPv6 address of both Ethernet interfaces by using the **ipv6 address** command. The configuration also enables router advertisement on both interfaces by using the **ipv6 nd prefix** command.

A default static route to router R2 is added. Note that the route points to the R2 link-local address. When a router uses a link-local address to the next-hop router, the output interface must be specified.

Tip These are long prefix lifetimes, although not as long as the defaults that are specified in the neighbor discovery RFC (a valid lifetime of 30 days and a preferred lifetime of 7 days). The selected lifetimes and the router advertisement interval (the interval between unsolicited router advertisements) are related. If the router sends a router advertisement every 10 minutes (for example), there is little reason to have such long prefix lifetimes. The value “infinity” is supported for deployments in which the prefixes should be considered valid regardless of the health of the subnet routers.

Cisco IOS show Commands

This topic describes two useful Cisco IOS Software commands for troubleshooting IPv6 issues.

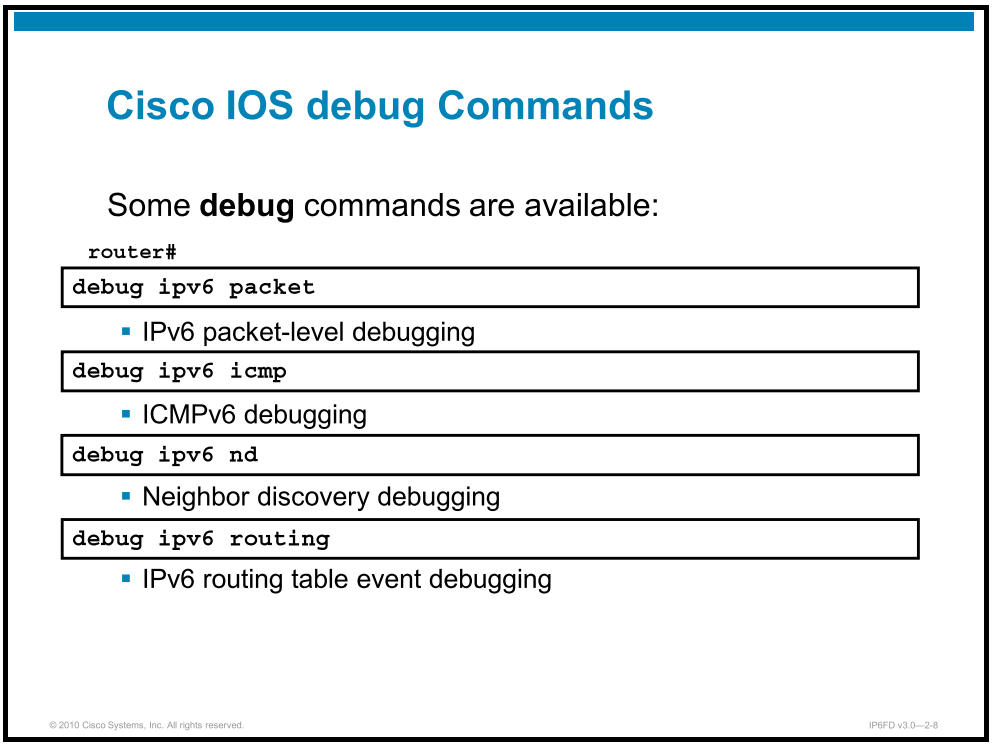


These two useful diagnostic commands are shown in the figure:

- **ping:** Use the **ping** command to validate IPv6 connectivity with the default router.
- **show ipv6 neighbors:** Use the **show ipv6 neighbors** command to examine the neighbor reachability cache on the router.

Cisco IOS debug Commands

This topic describes some useful Cisco IOS Software IPv6 **debug** commands.

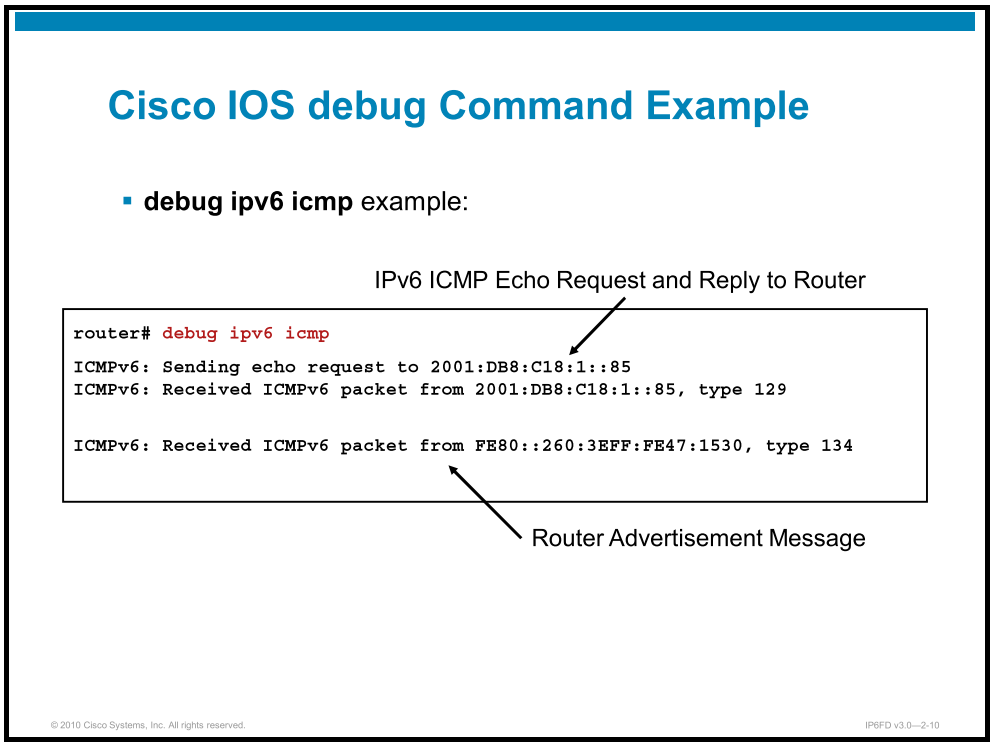


You can activate some debugging modes to examine IPv6 activity. When activated, these modes display information on the console.

- **debug ipv6 packet:** Enables IPv6 packet-level debugging
- **debug ipv6 icmp:** Enables ICMPv6 debugging, except neighbor discovery
- **debug ipv6 nd:** Enables ICMPv6 neighbor discovery debugging
- **debug ipv6 routing:** Enables IPv6 routing table event activity

Cisco IOS debug Command Example

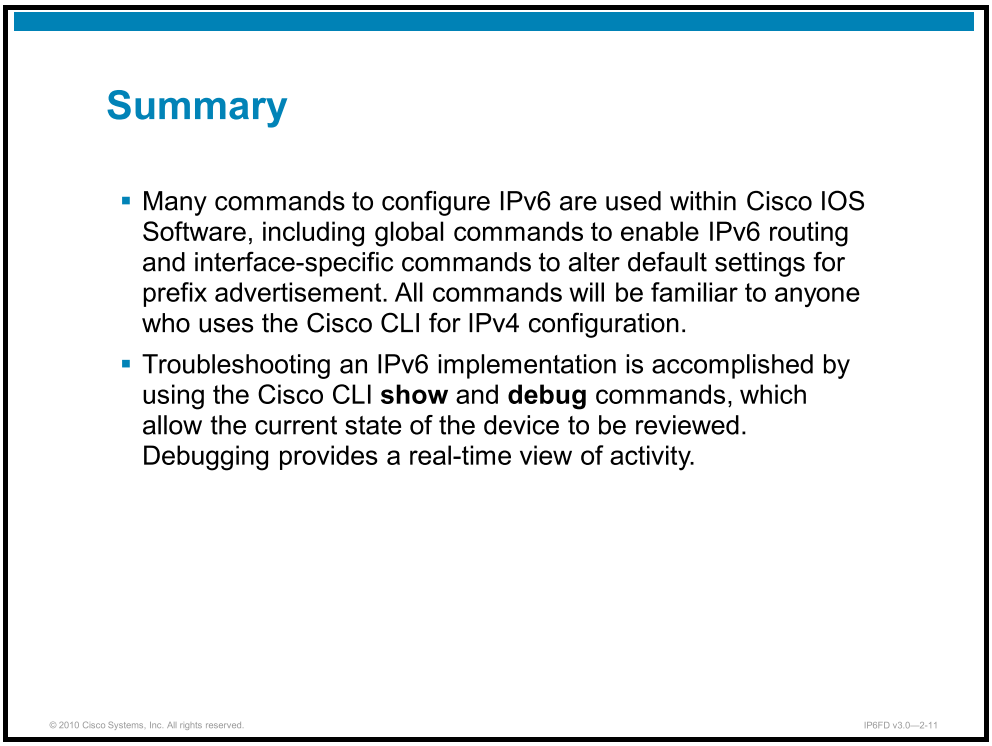
This topic describes sample output from the **debug ipv6 icmp** Cisco IOS Software command.



The figure shows an example in which IPv6 Internet Control Message Protocol (ICMP) debugging is activated and logs are received on the console port. The router sent an IPv6 ICMP echo request to a host, and an echo reply (type 129) was received. Similarly, a router advertisement was received on the link (type 134).

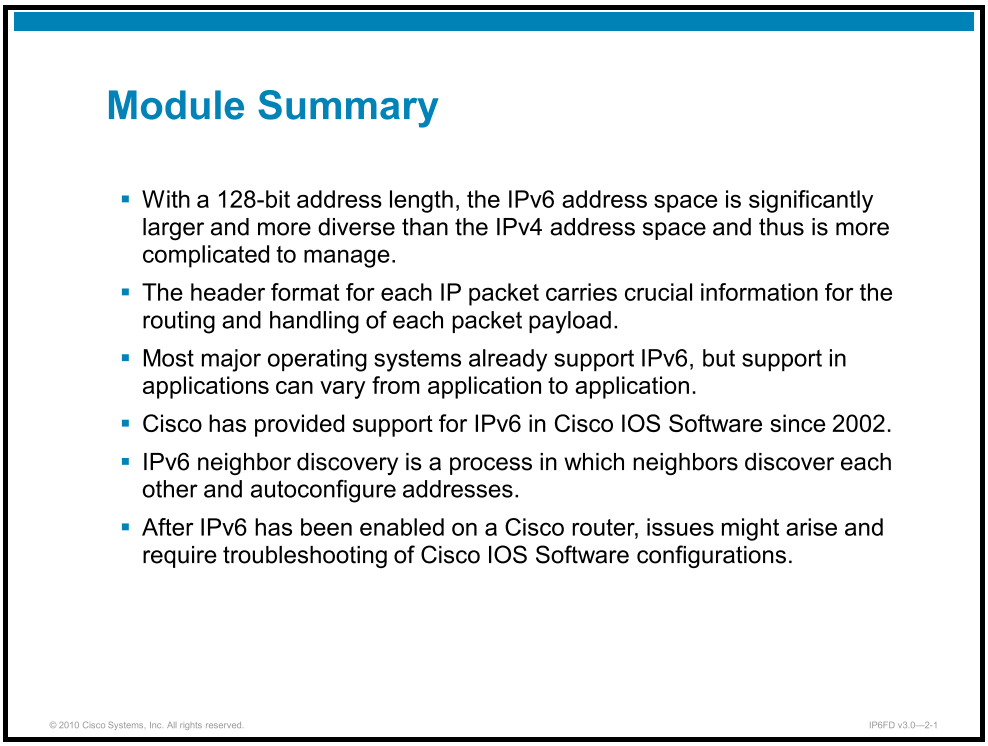
Summary

This topic summarizes the key points that were discussed in this lesson.



Module Summary

This topic summarizes the key points that were discussed in this module.



Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) Which address representation is incorrect? (Source: “Understanding the IPv6 Addressing Architecture”)

A) FE84:5:6:7::8
B) 2001:1:2::34
C) FFG2::1
D) FEC0:A:B:C::D

Q2) Anycast addresses exhibit which characteristics? (Source: “Understanding the IPv6 Addressing Architecture”)

A) They have only site-local scope, which uses unique local addresses.
B) They have properties such as multicast addresses.
C) They are indistinguishable from unicast addresses.
D) All addresses are anycast addresses but must be used with link-scope multicast mechanisms.

Q3) The IETF has allocated 2000::/3 to IANA for global unicast assignments. Which allocation is within that /3 prefix? (Source: “Understanding the IPv6 Addressing Architecture”)

A) 300A::/16
B) 4ABC:4367::/32
C) 0001::/16
D) 2001:DB8::/32

Q4) Which three IPv4 header fields were dropped when IPv6 was created? (Choose three.) (Source: “Describing the IPv6 Header Format”)

A) IHL
B) Type of Service
C) Header Checksum
D) Flags
E) Flow Label

Q5) The extension headers serve which important function in IPv6 networks? (Source: “Describing the IPv6 Header Format”)

A) identify optional processes that can be run on each IPv6 packet
B) allow IPv6 nodes to manipulate routers
C) identify processes that manipulate the routers in the path of a packet
D) replace the traditional role of TCP and UDP in a network

Q6) Which command enables IPv6 on a Cisco router? (Source: “Enabling IPv6 on Cisco Routers”)

A) **ipv6 routing enable**
B) **ipv6 unicast routing enable**
C) **ipv6 unicast-routing**
D) **ipv6 enable**

Q7) Match the IPv6 address assignment or enabling command with the appropriate classification. (Source: “Enabling IPv6 on Cisco Routers”)

A) **ipv6 address fe80::500 link-local**
B) **ipv6 address 2001:db8:7:8::/64 eui-64**
C) **ipv6 enable**
D) **ipv6 address 2001:db8:7:8::/64 link-local**

1. assigns a link-local address
2. enables IPv6 on an interface, resulting in an automatically generated link-local address
3. assigns a global-scope address, allowing the low 64-bit interface identifier to be generated automatically
4. illegal command that attempts to assign a global-scope address as a link-local address

Q8) Neighbor discovery is used for which three of these functions? (Choose three.) (Source: “Using ICMPv6 and Neighbor Discovery”)

A) discover Layer 2 addresses of on-link IPv6 peers
B) discover the Layer 2 address of the default router on the subnet
C) discover on-link routers that can act as default routers
D) discover the link-local IPv6 addresses of on-link neighbors

Q9) Stateless autoconfiguration relies on which mechanism to convey prefix information to hosts within the environment? (Source: “Using ICMPv6 and Neighbor Discovery”)

A) DHCPv6
B) multiprotocol DHCP
C) router advertisements that are sent by the subnet-local router
D) directed broadcasts that are sent by using UDP address updates

Q10) Which statement best describes the Redirect message functionality? (Source: “Using ICMPv6 and Neighbor Discovery”)

A) Redirect messages are sent by hosts to routers, to indicate that traffic should be redirected to an alternate destination.
B) Redirect messages are sent by routers to hosts, to allow them to autoconfigure addresses that use the included prefix.
C) Redirect messages are UDP-based messages that modify the neighbor table in link-local hosts.
D) Redirect messages are sent by routers to link-local nodes, to update their routing entry for off-link destinations with a better first-hop router.

Q11) Which ICMP Destination Unreachable code is not authentic? (Source: “Using ICMPv6 and Neighbor Discovery”)

A) no route to destination
B) port unreachable
C) malformed packet rejected
D) host unreachable

Q12) The Cisco IOS Software **show ipv6 neighbors** command shows which information? (Source: “Troubleshooting IPv6”)

A) routers that have formed a routing protocol neighbor relationship, such as OSPFv3
B) IPv6-capable neighbors, as determined by the Cisco Discovery Protocol at Layer 2
C) off-link nodes with which the local node has an established session
D) the neighbor cache contents, which show link-local node MAC address-to-IPv6 address mapping

Q13) What does the **debug ipv6 icmp** command show? (Source: “Troubleshooting IPv6”)

A) logging-buffer contents that relate to ICMPv6 errors
B) real-time ICMPv6 traffic, except that related to neighbor discovery
C) real-time ICMPv6 traffic for message router solicitation and router advertisement
D) RIPng messages that ride on the ICMPv6 protocol

Module Self-Check Answer Key

Q1) B

Q2) C

Q3) A

Q4) A, C, D

Q5) A

Q6) C

Q7) 1-A
2-C
3-B
4-D

Q8) A, B, C

Q9) C

Q10) D

Q11) C

Q12) D

Q13) B

Module 3

IPv6 Services

Overview

Diverse requirements and leveraging the network for business purposes continually compel the development of technologies to enhance network performance or increase network capabilities. Services such as quality of service (QoS) and multicasting, plus network management tools such as Secure Shell (SSH), FTP, Telnet, and ping are requirements for networks. This module describes the changes that are made to these tools and services to support IPv6 operations.

Module Objectives

Upon completing this module, you will be able to implement IPv6 services and applications. This ability includes being able to meet these objectives:

- Describe Mobile IPv6, and discuss the emerging technologies for network mobility and networking
- Describe IPv6 DNS in operation and IPv6 DNS client interactions
- Describe how DHCPv6 operates
- Describe how QoS is supported in IPv6
- Describe Cisco IOS tools, such as Telnet, TFTP, SSH, and others

Lesson 1

IPv6 Mobility

Overview

Traditionally, networks are stable entities that maintain semi permanent connections to the Internet and other networks. Increased reliance on IP-based communication solutions and a highly mobile society have generated a need for mobile networks—entire networks that may detach themselves from one point on the Internet and reattach themselves at another location. Network mobility will be far more common as IPv6 adoption increases. This lesson outlines IP mobility in general and describes the IPv6 Network Mobility model with possible usages.

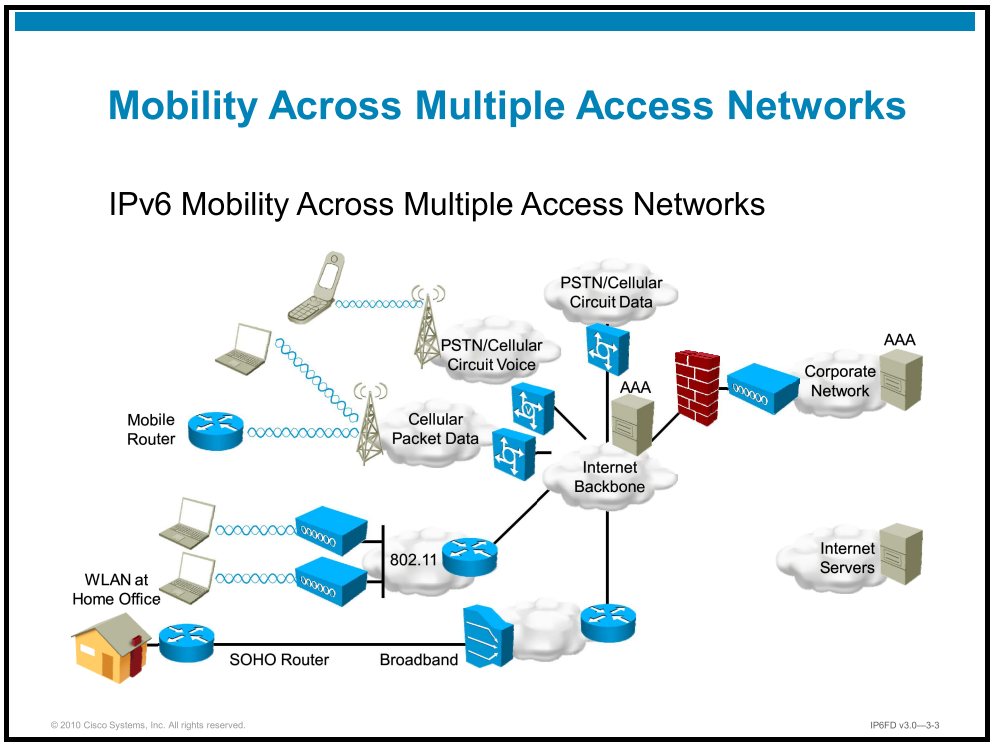
Objectives

Upon completing this lesson, you will be able to describe the Mobile IP model in an IPv6 environment. This knowledge includes being able to meet these objectives:

- Describe IP Mobility technology and issues
- Explore the Mobile IPv6 processes
- List the examples able to support or use Mobile IPv6

Introduction to IP Mobility

This topic describes the IP Mobility technologies.

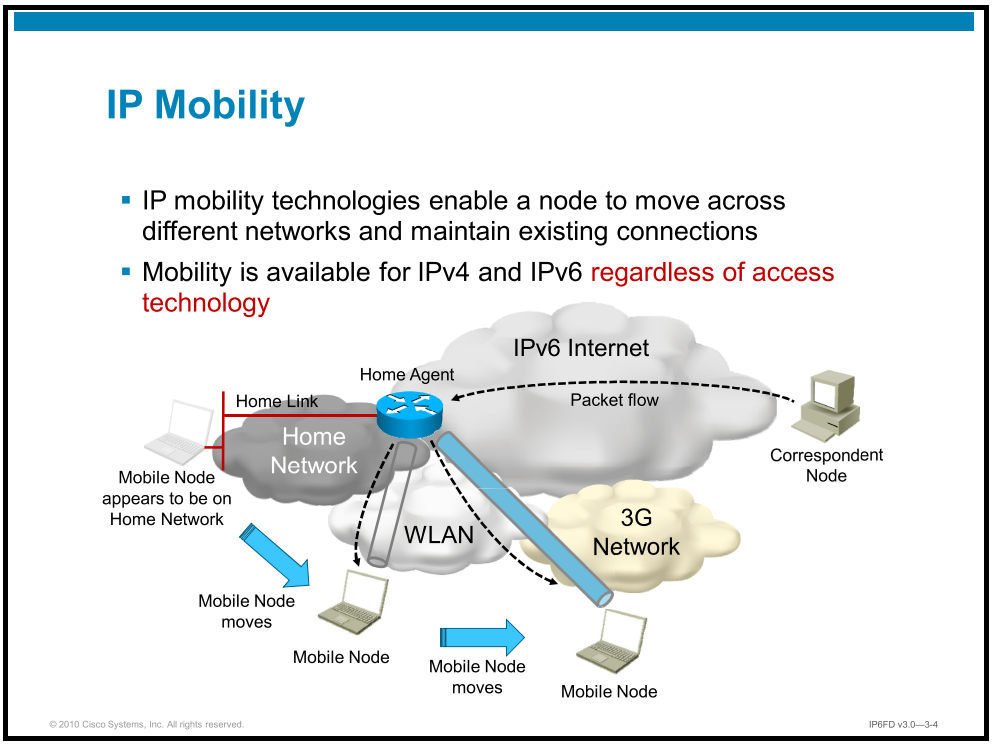


IP layer mobility is the only solution for devices that cross different access networks. The devices change their point of attachment in various access networks, but the goal is that the device is always reachable at its original IP address.

To push this scenario even further, a device should be able to maintain existing connections when moving from one access network to another, and this change should be transparent both to the user and to other nodes which this device communicates to.

The vast IPv6 address space enables easy deployment in the corporate and home networks.

An example would be that a car navigation unit would syncronize maps while being parked in the garage through the home WLAN, and continue doing so across the mobile network (3G/4G, etc.) when the car is being driven.



The IP mobility model is based on the concepts of home agents and mobile nodes. Nodes that move from their home networks cooperate with a mobility device, on their home networks, that is designated to assist in the Mobile IP process. These mobile nodes receive packets via this designated home agent.

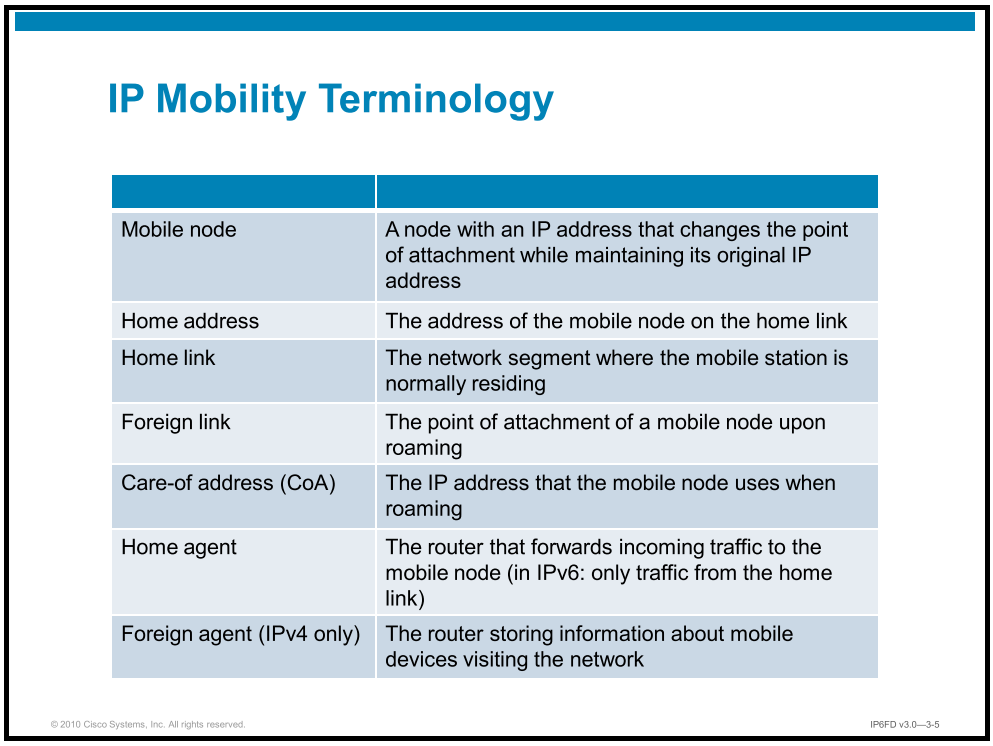
Packets that are sent to the mobile node on home link are always routed across the Internet to the home link of the node. If this node moves to another link (with another prefix), it would not receive those packets because packets addressed to the node are still routed to the router with the home link.

When a node changes its attachment point to the network, only the location of that node changes; the node identity does not. Because the IP address cannot distinguish between a node location and identity, Mobile IP is required.

Upon moving, the mobile node will register to the home agent with its new location and point of attachment. The home agent will then form a tunnel to the mobile node, sending the data with the mobile node as the destination. The home agent will also represent the mobile node on the link, forwarding the traffic from the home link to the mobile node.

Note	Mobile IPv6 is essentially a tunnel broker, with one end of the tunnel being stable, and the other one moving.

Reference	Mobile IPv4 is defined in RFC 3344, *IP Mobility Support for IPv4.* Mobile IPv6 is described in RFC 3775, *Mobility Support in IPv6.*



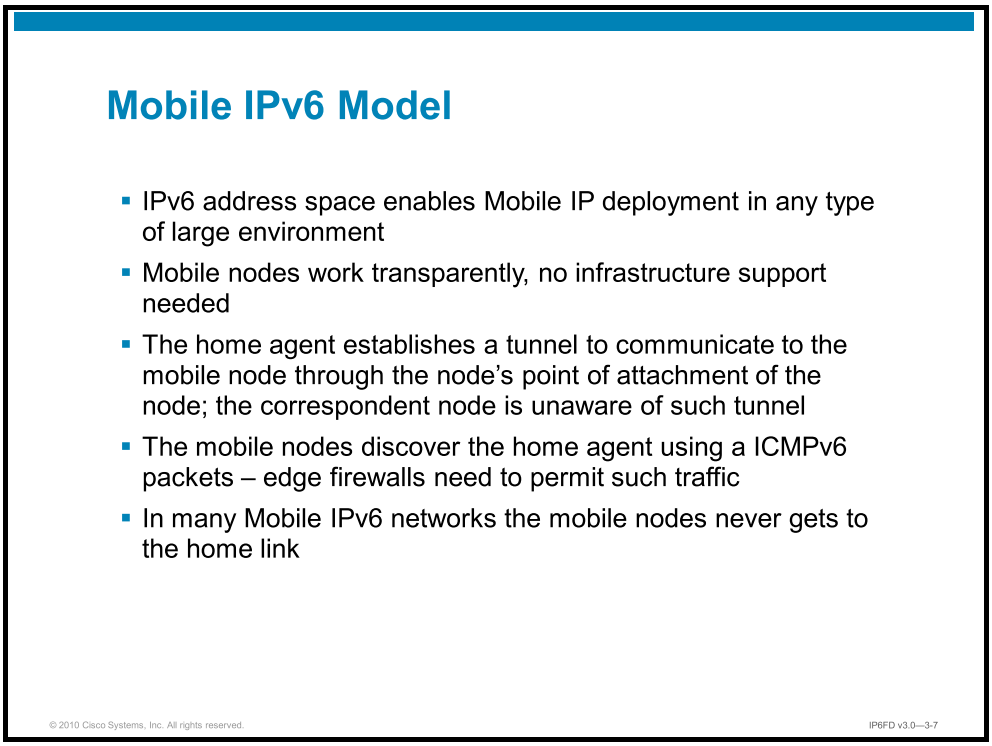
The elements of Mobile IP networks are:

- **Mobile Node:** A node that can change its point of attachment from one link to another, while still being reachable via its home address.
- **Home Address:** A unicast routable address that is assigned to a mobile node and is used as the permanent address of the mobile node. The address is within the home link of the mobile node. Standard IP routing mechanisms will deliver packets that are destined for the home address of a mobile node to its home link. Mobile nodes can have multiple home addresses, for instance, when there are multiple home prefixes on the home link.
- **Home Link:** The home subnet prefix of a mobile node is defined on the home link.
- **Foreign Link:** Any link other than the home link of the mobile node.
- **Care-of Address:** A unicast routable address that is associated with a mobile node while visiting a foreign link. The subnet prefix of this IP address is a foreign subnet prefix. Among the multiple care-of addresses that a mobile node may have (for example, with different subnet prefixes), the address that is registered with the home agent of the mobile node, for a given home address, is called its "primary" care-of address.
- **Home Agent**: A router on the home link of a mobile node, with which the mobile node has registered its current care-of address. While the mobile node is away from home, the home agent intercepts packets on the home link that are destined for the home address of the mobile node. The home agent then encapsulates the packets and tunnels them to the registered care-of address of the mobile node.
- **Foreign Agent:** A router storing information about mobile devices visiting the network. It is also used to terminate the tunnels to various home agents.

Note Only Mobile IPv4 networks use the foreign agent.

Mobile IPv6

This topic describes the Mobile IPv6 model.



The IPv6 mobility model has the following characteristics:

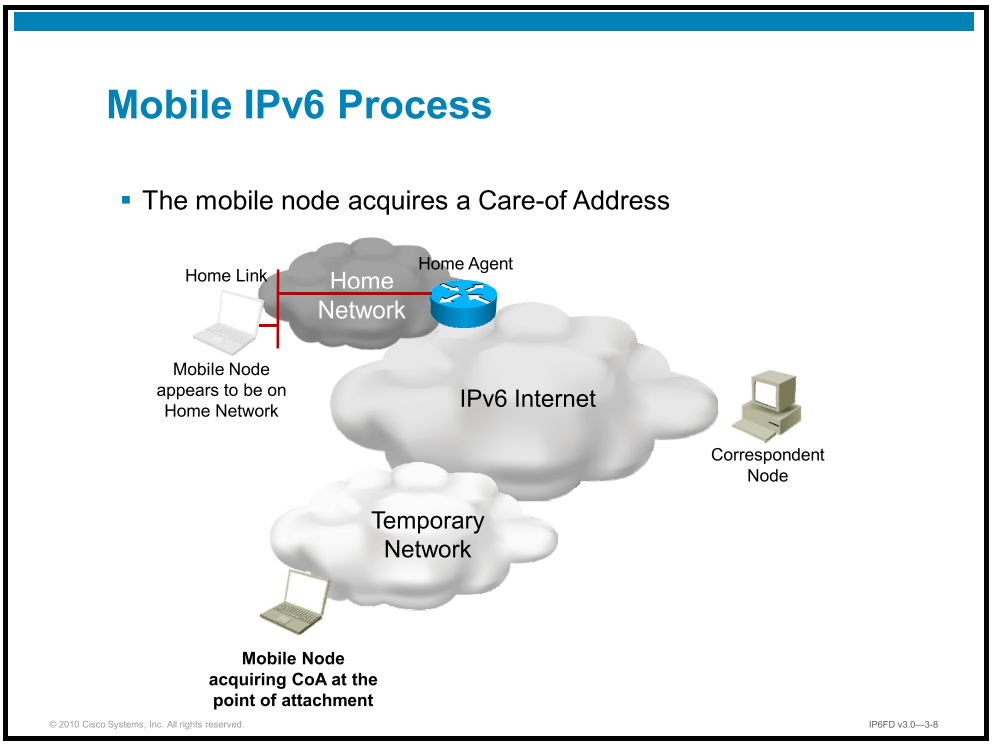
- IPv6 address space enables Mobile IP deployment in any kind of large environment. The available address space is relatively large and is able to accommodate a large number of devices in any /64 subnet.
- The Mobile IPv6 model takes advantage of the benefits of the IPv6 protocol itself. Examples include option headers, Neighbor Discovery, and autoconfiguration. These additions are all included in standard IPv6 stacks to which network devices adhere to, so no equipment upgrades are necessary.
- The home agent communicates with the mobile node by establishing a tunnel. This tunnel is used to convey the packets from the correspondents that communicate to the mobile node. Since the correspondent nodes use the original prefix as a destination address for the mobile node, the packets are routed to the home agent first, which needs to forward them to the actual point of attachment of the mobile node.

Note	In some cases, routing through the home agent is eliminated because Mobile IPv6 route optimization allows mobile nodes and corresponding nodes to communicate directly. Support for route optimization is a fundamental part of the protocol, rather than a nonstandard set of extensions.

- Mobile nodes use a special protocol to discover the home agent while they roam on a foreign link. The discovery process uses ICMPv6 packets, so it is essential that they are permitted to pass through firewalls on their way.

Note	While away from home in Mobile IPv6, some packets are sent to a mobile node using an IPv6 routing header rather than IP encapsulation. Using an IPv6 routing header reduces the amount of resulting overhead compared to Mobile IPv4.

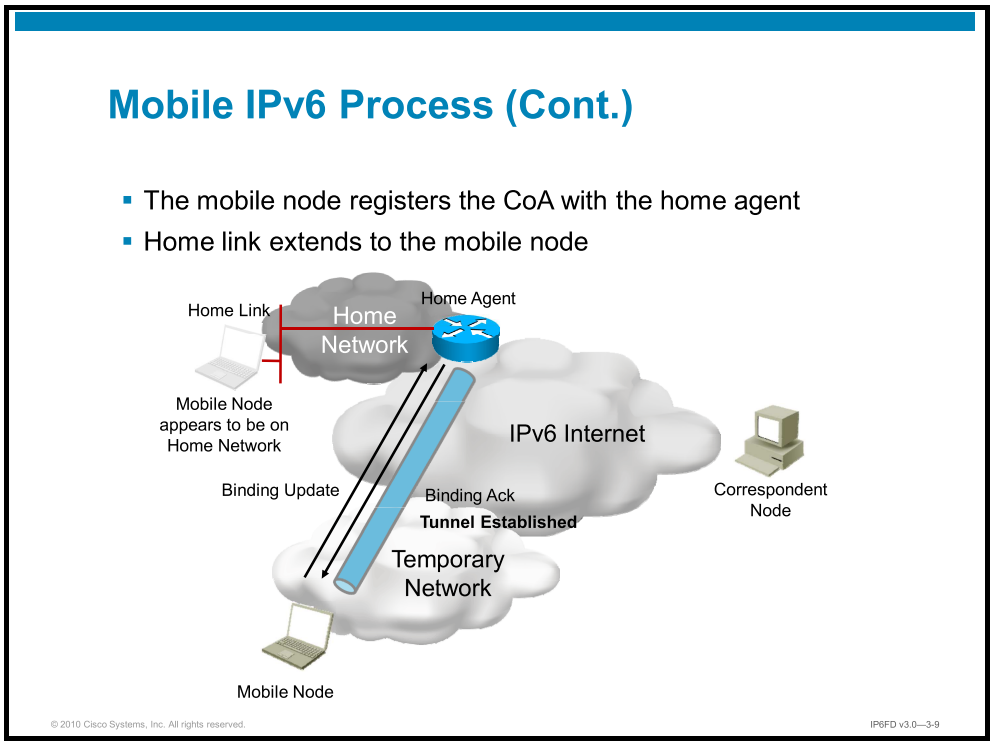
- Many solutions that use Mobile IPv6 predict that the mobile node will never appear on its actual home link in the life of the device. For example, a mobile phone using MIPv6 will never find itself directly attached to the network segment of the mobile operator local network infrastructure.



The following process describes the operation of Mobile IPv6:

Stage 1: CoA is acquired. The Mobile IPv6 node obtains an IPv6 Care-of Address (CoA) in the visited network through stateless or stateful autoconfiguration.

A CoA is an IP address that is associated with a mobile node that has the subnet prefix of a particular foreign link. The mobile node can acquire its CoA through conventional IPv6 mechanisms, such as stateless or stateful autoconfiguration.

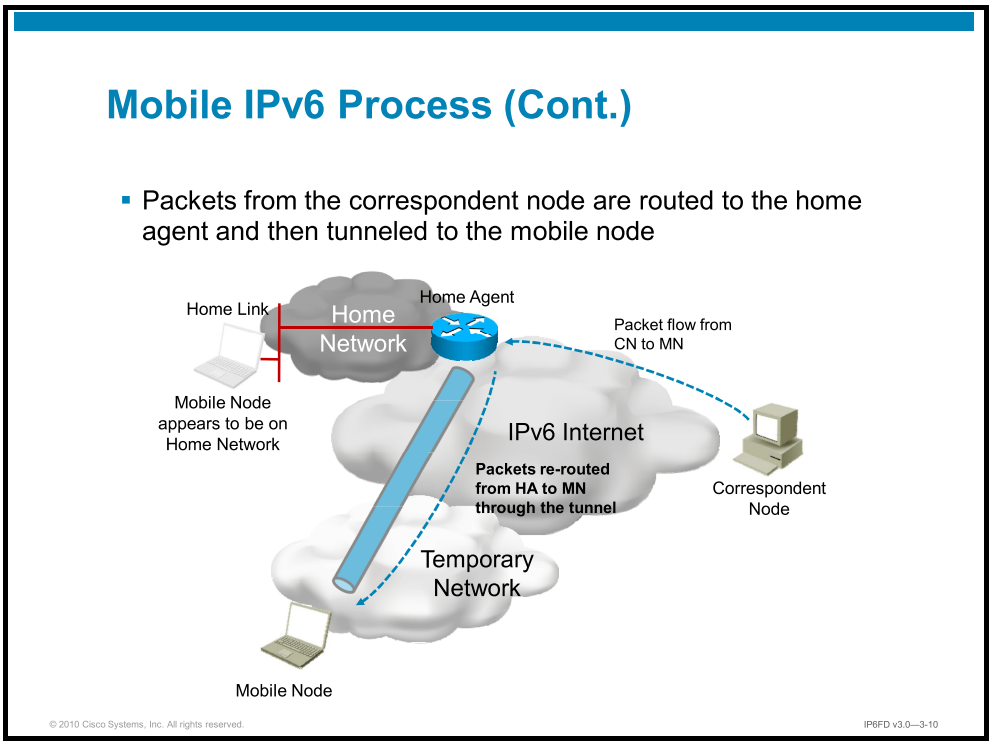


Stage 2: Primary CoA is registered. The association between the home address of a mobile node and CoA is known as a "binding" for the mobile node. The mobile node uses a Mobile IPv6 binding update to register its primary CoA with a router on its home link, requesting that this router function as the home agent of the mobile node. The home agent will acknowledge the binding and establish an IPv6-in-IPv6 tunnel to the mobile node.

Mobile IPv6 also provides support for multiple home agents and limited support for the reconfiguration of the home network. In these cases, the mobile node may not know the IP address of its own home agent and even the home subnet prefixes may change over time.

Known as "dynamic home agent address discovery," this mechanism allows a mobile node to dynamically discover the IP address of a home agent on its home link, even when the mobile node is away from home. Mobile nodes can also learn new information about home subnet prefixes through the mobile prefix discovery mechanism.

Note The home link is extended to the mobile node because the mobile node needs to appear to its neighboring devices on the local link as locally attached. The home link is now "elastic".



There are two possible modes for communications between the mobile node and a correspondent node:

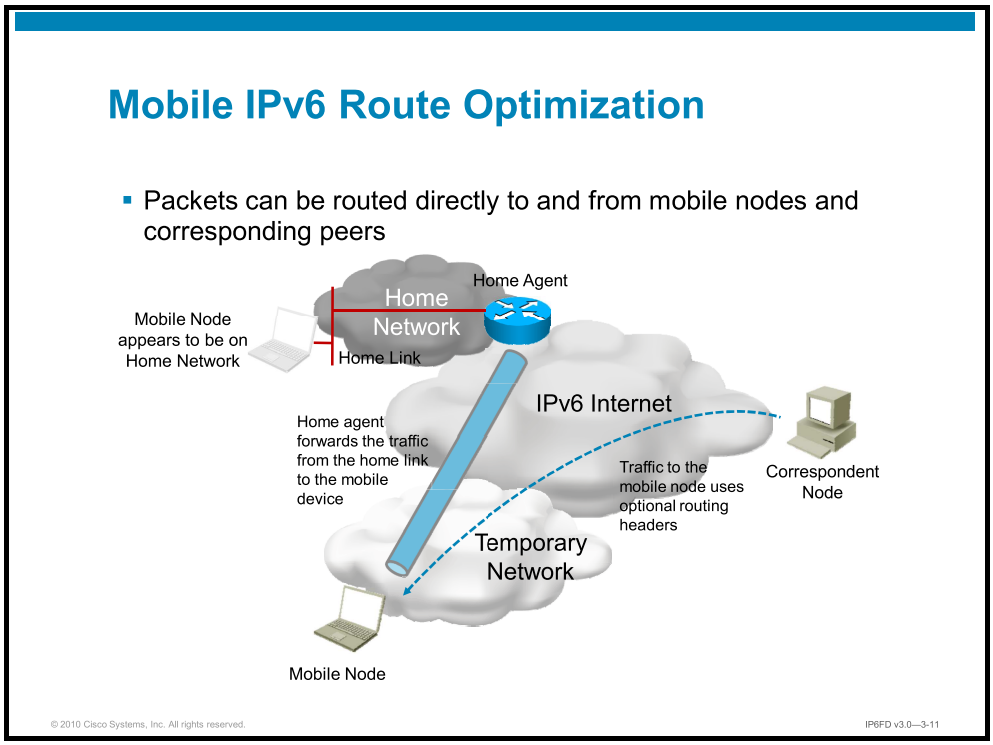
- Bidirectional tunneling
- Route optimization

Stage 3: Packets from the correspondent node are routed to the home agent. When using *bidirectional tunneling*, Mobile IPv6 support is not required on the correspondent node. Packets from the correspondent node are routed to the home agent and then tunneled to the mobile node.

In this mode, the home agent uses proxy neighbor discovery to intercept any IPv6 packets addressed to the home address (or home addresses) of the mobile node on the home link. Each intercepted packet is tunneled to the primary CoA of the mobile node. This tunneling is performed using IPv6 encapsulation.

Stage 4: Packets are tunneled to the mobile node. Packets to the correspondent node are reverse-tunneled from the mobile node to the home agent. There they are routed normally from the home network to the correspondent node.

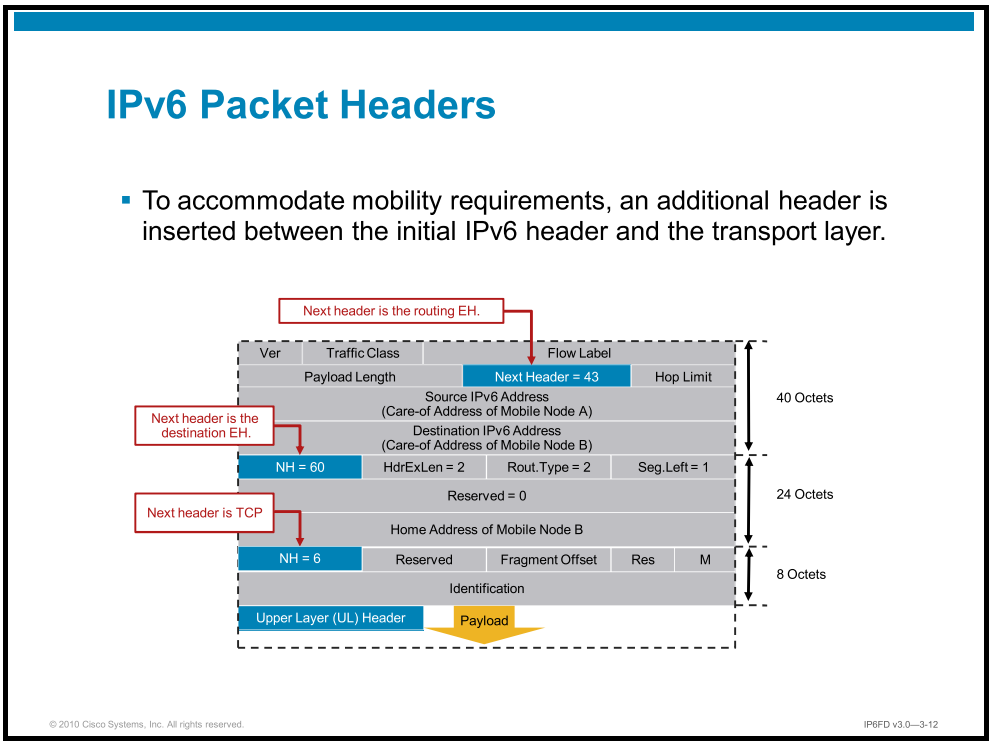
Note It is important to note that this kind of tunneling has nothing to do with IPv6 transition. No IPv4 is involved in Mobile IPv6—it is an “all IPv6” service.



When using *route optimization* the mobile node needs to register its current binding at the correspondent node. Packets from the correspondent node can be routed directly to the CoA of the mobile node. When sending a packet to any IPv6 destination, the correspondent node checks its cached bindings for an entry for the destination address of the packet. If a cached binding for this destination address is found, the node uses a new type of IPv6 routing header to route the packet to the mobile node by way of the CoA indicated in this binding.

Routing packets directly to the CoA of the mobile node allows the shortest communications path to be used. It also eliminates congestion at the home agent of the mobile node and home link. In addition, the impact of any possible failure of the home agent or networks on the path to or from it is reduced.

Note Route optimization is part of Mobile IPv6, but is not necessarily supported or used in all IPv6 stack implementations.

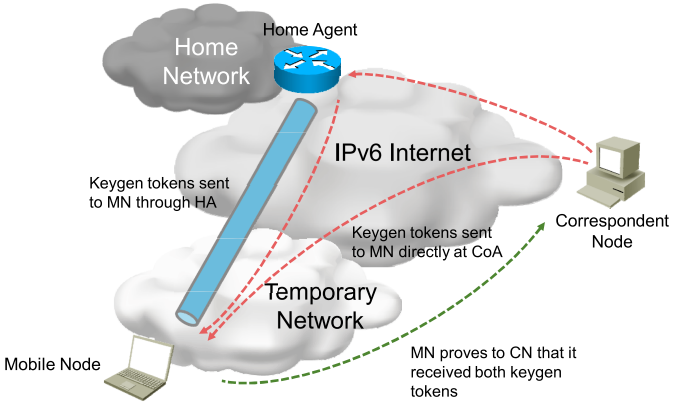


When routing packets directly to the mobile node, the correspondent node sets the destination address in the IPv6 header to the CoA of the mobile node. A new type of IPv6 routing header is also added to the packet to carry the desired home address. Similarly, the mobile node sets the source address in the IPv6 header of the packet to its current CoAs. The mobile node adds a new IPv6 home address destination option to carry its home address. The inclusion of home addresses in these packets makes using the CoA transparent above the network layer (for example, at the transport layer).

Note One advantage of Mobile IPv6 is masking the current CoA for an away-from-home node. When you are not using route optimization—regardless of the CoA of the mobile node—the mobile node appears to be at home to correspondent nodes. Only the home agent knows that the mobile node is away from home. If, however, the mobile node attempts to use the correspondent node binding feature of Mobile IPv6 support, that correspondent node will be aware that the mobile node is no longer on the home network. If this awareness is considered a security issue, route optimization can and should be disabled.

Return Routability Procedure

- Assurance that the mobile node is addressable at its claimed CoA, and also at its home address



- Once the identity is verified, an IPsec secured connection can be established end-to-end; no need to encrypt signaling

Traffic goes through the home agent until the return routability procedure is performed. Signaling is completed via the home agent, and home registrations keep the home agent informed.

To use the return routability procedure, the correspondent node must support Mobile IPv6 and requires a mobile network to register its binding association to the correspondent node. The mobile network can also be a correspondent node to communicate with other mobile networks.

The return routability procedure enables the correspondent node to obtain some reasonable assurance that the mobile node is, in fact, addressable at its claimed CoA and also at its home address. Only with this assurance can the correspondent node accept binding updates from the mobile node. The mobile node then instructs the correspondent node to direct the data traffic of that mobile node to its claimed CoA.

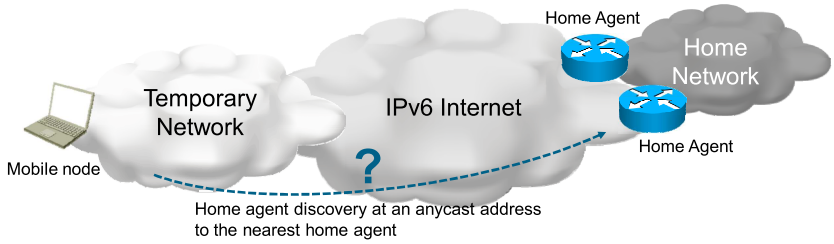
This procedure tests whether packets that are addressed to the two claimed addresses are routed to the mobile node. The mobile node can pass the test only if it is able to supply proof that it received certain data (the keygen tokens), which the correspondent node sends to those addresses.

Once the identity of the mobile node is verified, the correspondent node can use an IPsec encrypted session to communicate with the mobile node securely (and vice-versa). There is no need to encrypt signaling traffic (e.g. binding updates, discovery messages, etc.).

Note RFC 4449, *Securing Mobile IPv6 Route Optimization Using a Static Shared Key*, improves the return routability procedure to protect mobile node-correspondent node bindings, at the expense of requiring additional in-advance setup. In this method, the two parties share a secret key that establishes initial trust.

Dynamic Home Agent Address Discovery

- Mobile nodes may discover home agents on their home link by using a home agent address discovery request.
- The request is sent to the anycast address of the Mobile IPv6 home agent.
- The home agent responds with a home agent address discovery reply message.
- The mobile node may then complete home registration and receive services from that home agent.



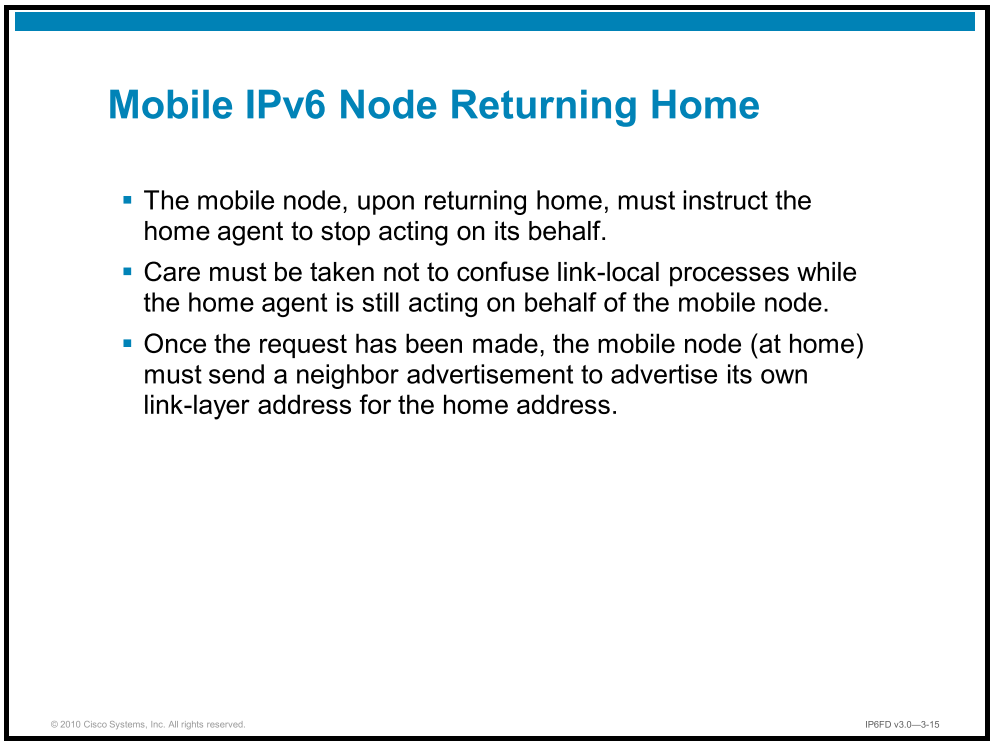
IP6FD v3.0—3-14

When the mobile node needs to send a binding update to its home agent to register its new primary CoA, the mobile node may not know the address of any router that can serve as a home agent on its home link. For example, some nodes on the home link of a mobile node may have been reconfigured while the mobile node was away from home. Therefore, a different router replaced the router that was operating as the home agent of the mobile node.

In this case, the mobile node may attempt to discover the address of a suitable home agent on its home link. To do so, the mobile node sends an ICMP home agent address discovery request message for its home subnet prefix to the anycast address of the Mobile IPv6 home agent (the subnet prefix, followed by all 1s except for the Universal and Local bit for EUI-64 addresses, and the last seven bits, which for this anycast address is 7E).

The home agent, on the home link that receives this request message, will return an ICMP home agent address discovery reply message. The message gives the addresses for the home agents operating on the home link. The mobile node, upon receiving this home agent address discovery reply message, may then send its home registration binding update to any of the unicast IP addresses listed in the home agent addresses field in the reply.

Note Dynamic home agent discovery, while a powerful solution for nodes that rarely return to the home network, can also expose certain security issues because these messages are unauthenticated. Suppose a vulnerability is discovered for a well-known and widely deployed home agent. Attackers could sweep the Internet posing as mobile nodes, attempting to connect to Mobile IPv6 home agent anycast addresses. By finding an active home network, they would be told the location of all the home agents. In a high-security deployment, this feature can be disabled.



A mobile node determines that it has returned to its home link through the movement detection algorithm when the mobile node detects that its home subnet prefix is again on-link. The mobile node should then send a binding update to its home agent instructing it to no longer intercept or tunnel packets for it. In this home registration, the mobile node must set the acknowledge (A) bit and the home registration (H) bit. It must also set the CoA for the binding to the home address of the mobile node. The mobile node must use its home address as the source address in the binding update. The mobile node sets the A and H bits as follows:

- The sending mobile node sets the A bit to request the return of a binding acknowledgment upon receipt of the binding update.
- The sending mobile node sets the H bit to request that the receiving node act as the home agent of this node. The destination of the packet carrying this message must be that of a router sharing the same subnet prefix as the home address of the mobile node in the binding.

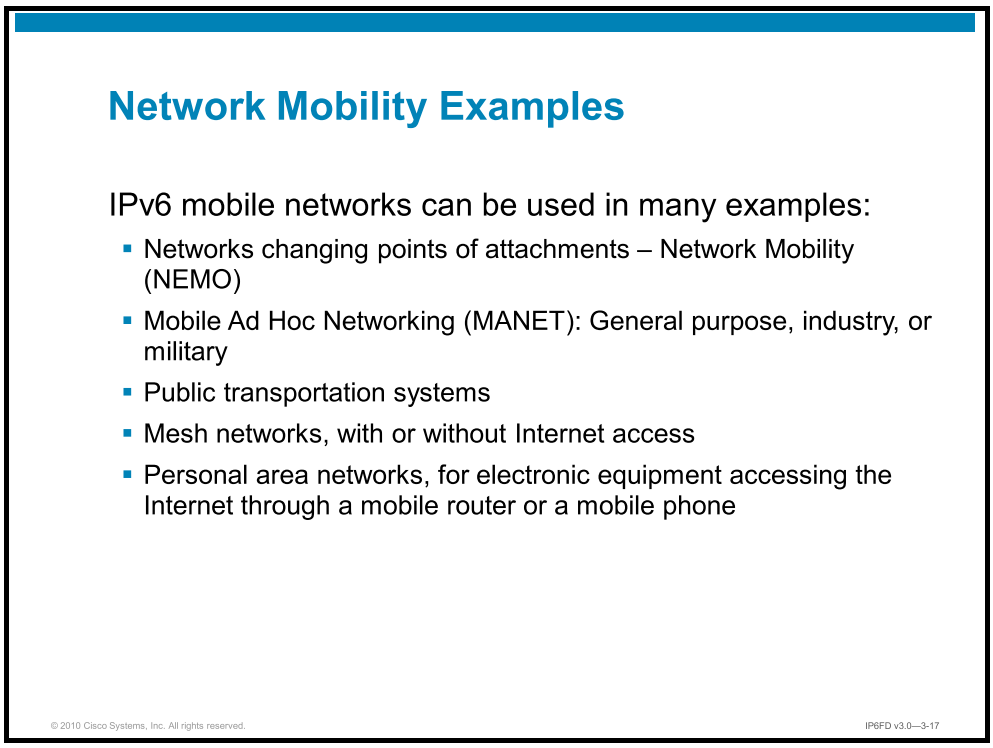
In this special case of the mobile node returning home, the mobile node must send a multicast packet and, in addition, set the source address of this neighbor solicitation to the unspecified address (0:0:0:0:0:0:0:0). The target of the neighbor solicitation must be set to the home address of the mobile node. The destination IP address must be set to the solicited-node multicast address of the home address of a mobile node.

The home agent will send a multicast neighbor advertisement back to the mobile node with the solicited (S) flag set to zero. The mobile node then sends its binding update to the MAC address of the home agent, instructing its home agent to no longer serve as a home agent for it. By processing this binding update, the home agent will cease defending the home address of the mobile node for Duplicate Address Detection and will no longer respond to neighbor solicitations for the home address of the mobile node. The mobile node is then the only node on the link receiving packets at the home address of the mobile node.

After the mobile node sends the binding update, it must be prepared to reply to neighbor solicitations for its home address. Such replies must be sent using a unicast neighbor advertisement to the MAC address of the sender. After receiving the binding acknowledgment for its binding update to its home agent, the mobile node must send a multicast packet onto the home link (to the all-nodes multicast address) to advertise the MAC address of the mobile node for its own home address.

Network Mobility Examples

This topic describes mobile network characteristics, requirements, and problems, and how IPv6 addresses them.

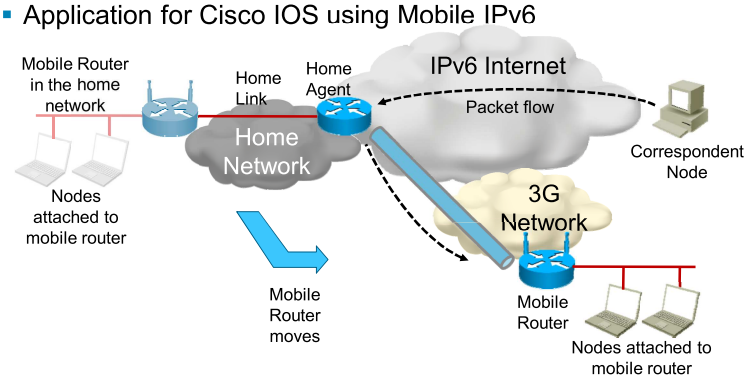


In the future, most IP devices will be mobile and will always be connected to the Internet by some means. Networks will also be mobile. Mobile networks may be of any size, ranging from a few IP devices to thousands of IP devices.

Mobile IP specifications do not provide explicit support of mobile networks. However, because mobile networks have specific characteristics, requirements, and problems, IP needs explicit support for mobile networks.

Network Mobility—NEMO

- A mobile router attaches a network to the IPv6 Internet while being able to change points of attachment
- Application for Cisco IOS using Mobile IPv6

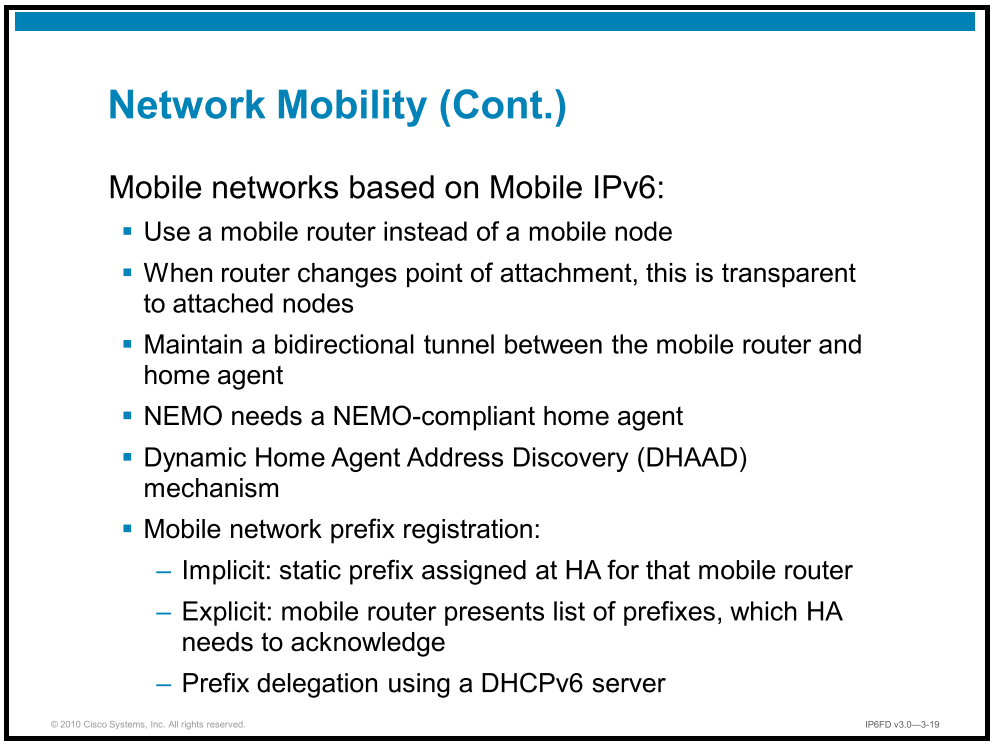


Communication sessions to mobile router-attached nodes continue uninterrupted

A mobile network is a network segment or subnet that can move and attach to arbitrary points on the Internet. A mobile network can be accessed only via specific gateways, called mobile routers, which manage its movement, as follows:

- Mobile networks have at least one mobile router serving them. A mobile router does not distribute mobile network routes to the infrastructure at its point of attachment, for example, in the visited network. Instead, it maintains a bidirectional tunnel to a home agent that advertises an aggregation of mobile networks to the infrastructure.
- The mobile router is also the default gateway for the mobile network.
- The mobile router advertises one or more prefixes in the mobile network that is attached to it. Most commonly, a default route is advertised to the nodes connected to the mobile router.

Note The Network Mobility (NEMO) is defined in RFC 3963, *Network Mobility (NEMO) Basic Support Protocol*. It is supported in Cisco IO Software.



The NEMO is an application for Cisco IOS which enables a router to move to any point in the IPv6 Internet and still be reachable from any point at its original IP address. NEMO extends the concept of Mobile IPv6 from a mobile node to a mobile router.

When the mobile router moves away from the home link and attaches to a new access router, it acquires a care-of address (CoA) from the visited link. Using the CoA, it immediately sends a binding update to its home agent. When the home agent receives this binding update, it creates a binding cache entry that binds the home address of the mobile router to its current CoA.

If the mobile router wishes to act as a mobile router and provide connectivity to nodes in the mobile network, it indicates this desire to the home agent by setting a router flag (R) in the binding update. It may also include information about the mobile network prefix in the binding update. The home agent can then forward packets that are meant for nodes in the mobile network to the mobile router. A new mobility header option is specified for mobile networks.

Note	All traffic between the nodes in the mobile network and correspondent nodes passes through the home agent.

A NEMO-compliant home agent can operate as a Mobile IPv6 home agent.

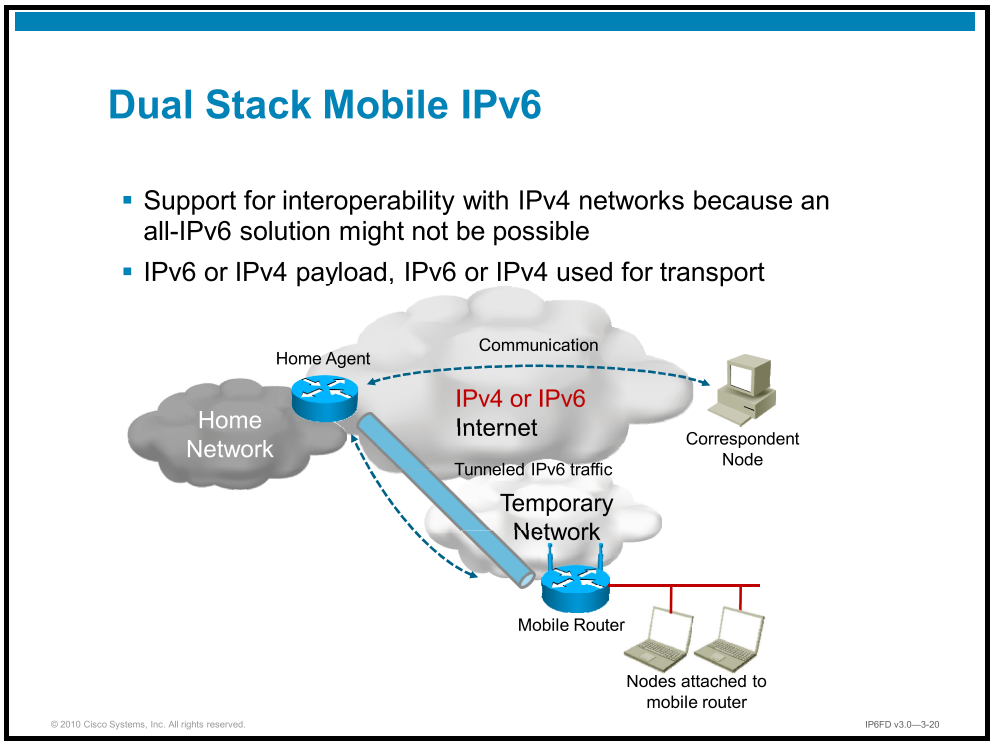
The dynamic home agent address discovery (DHAAD) mechanism allows a mobile node to discover the address of the home agent on its home link:

- The 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.

The mobile router will acquire the prefix for the mobile network and for the devices attached to it in one of the following ways:

- *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:* 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.
- *Prefix assignment using Prefix Delegation:* The prefix is acquired from the centrally located DHCPv6 server configured for prefix delegation. The mobile router will then use the assigned prefix obtained from DHCP, and adveritse it using Router Advertisements on the local link to auto-configure the nodes attached to the mobile router.



Mobile IPv6 and NEMO allow mobile nodes to move within the IPv6 Internet while maintaining reachability and ongoing sessions, using an IPv6 home address or prefix.

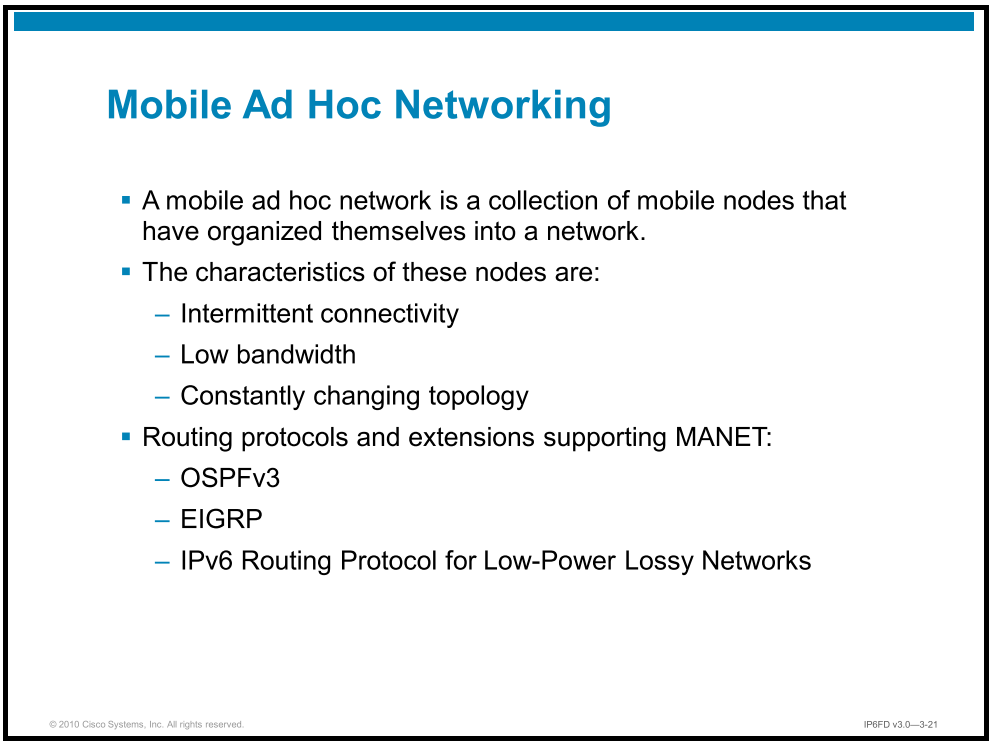
However, since IPv6 is not widely deployed, it is unlikely that mobile nodes will initially use only IPv6 addresses for their connections. It is reasonable to assume that mobile nodes will need an IPv4 home address that can be used by upper layers.

Mobile nodes will move to networks that might not support IPv6 and would therefore need the capability to support an IPv4 care-of address. Dual Stack Mobile IPv6 allows dual stack mobile nodes to request that their home agent (also dual stacked) tunnels IPv4/IPv6 packets addressed to their home addresses, as well as IPv4/IPv6 care-of address(es).

DSMIPv6 solves the following issues:

- Mobile nodes are able to use IPv4 and IPv6 home or care-of addresses simultaneously and update their home agents accordingly.
- Mobile nodes need to be able to know the IPv4 address of the home agent as well as its IPv6 address. There is no need for IPv4 prefix discovery.
- Mobile nodes need to be able to detect the presence of a NAT device and traverse it in order to communicate with the home agent.

Note Dual stack Mobile IPv6 is defined in the RFC 5555, *Mobile IPv6 Support for Dual Stack Hosts and Routers.*



The goal of mobile ad hoc networking is to provide robust and efficient operation in mobile wireless networks by incorporating routing functionality into the mobile nodes themselves. These networks have the following characteristics:

- **Dynamic, rapidly changing:** Nodes frequently leave and join.
- **Intermittent connectivity:** Nodes may periodically lose their connection to the rest of the network.
- **Low bandwidth:** The wireless technologies that are involved typically have very slow uplinks.
- **Short distance:** The wireless nodes may have limited range.

Initial applications for Mobile Ad Hoc Networking (MANET) involved the military and transportation sectors.

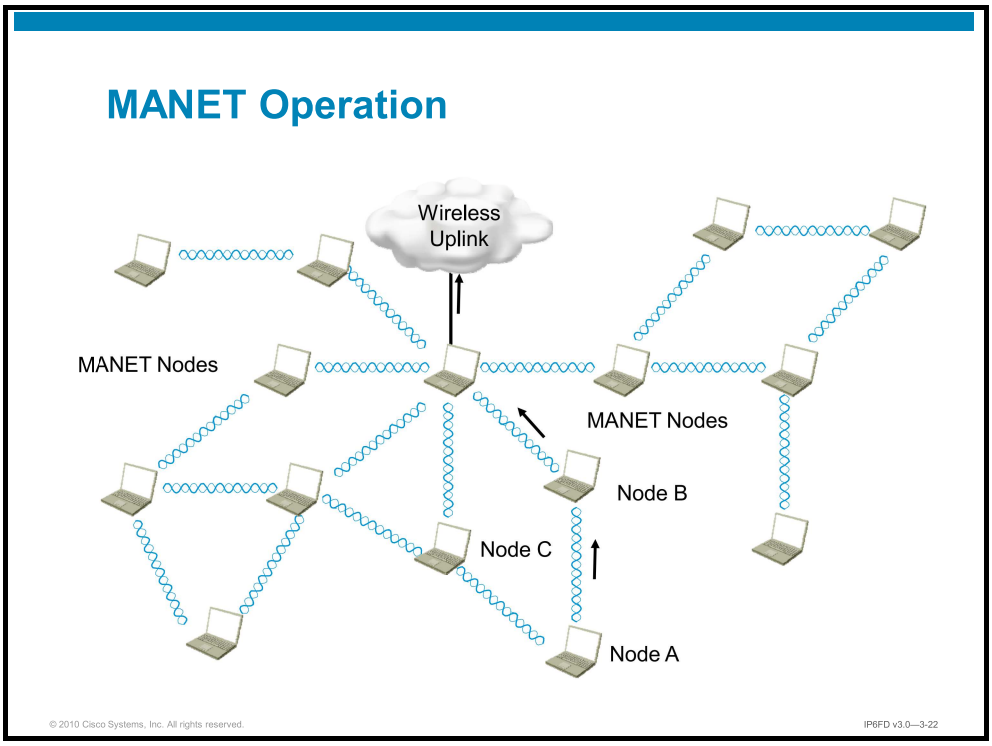
In the military arena, one potential application is to enable small wireless sensors with MANET for deployment in the battlefield. Once dispersed, the sensors would organize themselves into a network to exchange data and information on how to reach the network uplink. Because of the hostile environment, these sensors are expected to be destroyed, moved, or otherwise have their connections to the rest of the sensors disrupted on a regular basis.

In the transportation field, MANET is being considered as a way to dynamically update vehicles regarding traffic conditions. Vehicles near a traffic disruption would alert other vehicles with MANET regarding current traffic conditions. This information would propagate its way from car to car, enabling drivers not yet affected by a traffic incident to select alternative routes to avoid the disruption.

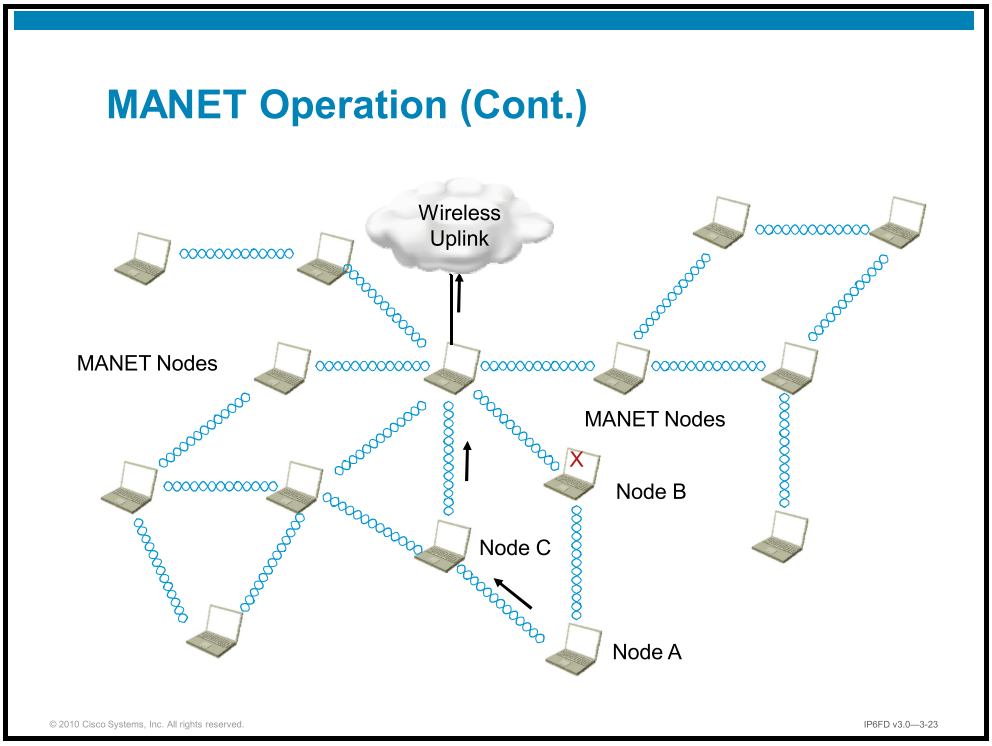
Routing protocols supporting MANET operation include OSPFv3, EIGRP, and IPv6 Routing Protocol for Low-Power Lossy Networks (RPL for LLNs). The key issues are rapidly changing topology (when devices arrive and leave), end link metric reporting when conditions change. Routing protocols feature an interface with the radio system of the device, and react to link quality changes.

Low-power lossy networks consist of devices limited with processing power and radio range, but very small in physical size. Due to their simplicity, such devices can be deployed in very large numbers and organize themselves into a network. Examples would include “sensor dust”, etc.

Note OSPFv3 and EIGRP extensions to support MANET are implemented in Cisco IOS; the IPv6 Routing Protocol for Low-power Lossy Networks is currently in the phase of an IETF draft.



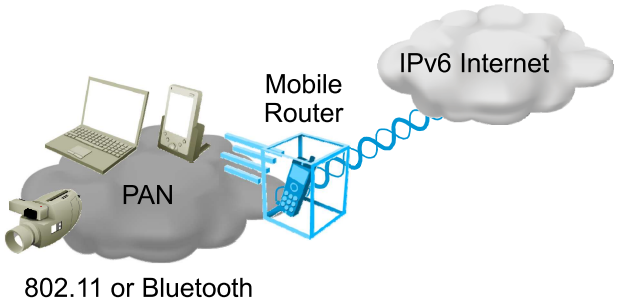
In the example, a collection of MANET-enabled mobile wireless nodes have organized themselves into a dynamic network to exchange information. Node A, at the bottom of the picture, has wireless links to two other MANET Nodes, B and C. Through execution of MANET routing protocols, Node A chooses the link to Node B to provide access to the rest of the MANET network.



When Node B loses connectivity with the network, Node A must reapply its MANET routing protocols to determine a new path to the rest of the network. It chooses its connection to Node C, and communication with the rest of the nodes is re-established.

IPv6 Personal Area Networks

- A mobile router is a personal communication device that is acting as a router and serving Internet access to peripherals.
- Digital cameras, PDAs, laptops, headsets, etc., use the mobile router for Internet access.
- These devices can use any mobile router to connect to the Internet.

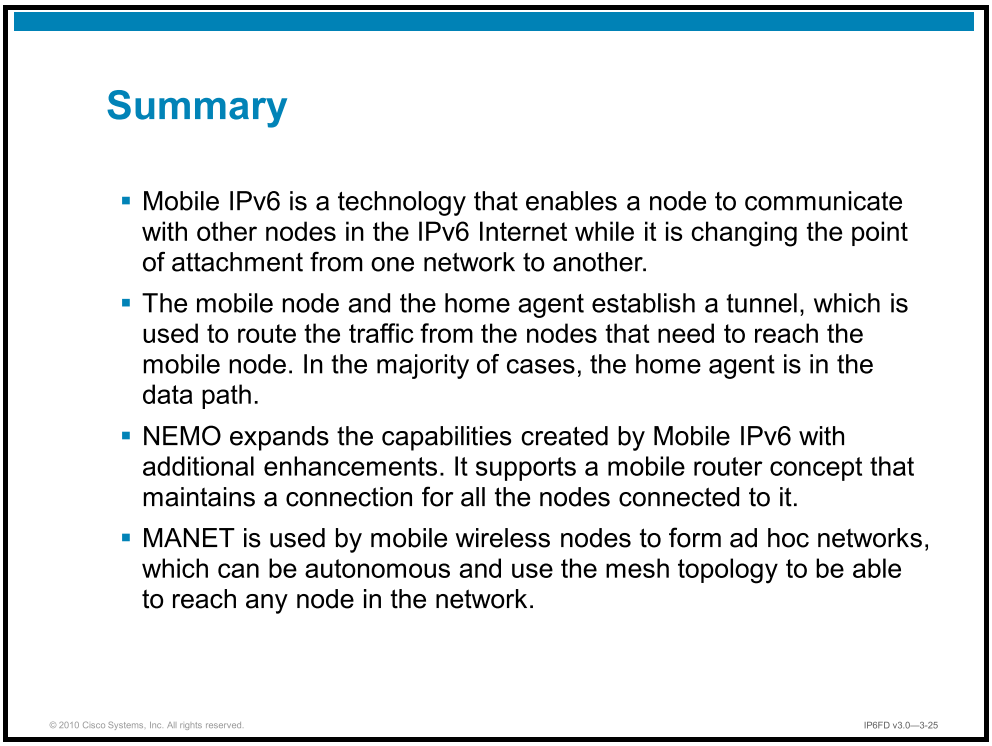


Accelerated by the success of cellular technologies, mobility has changed the way people communicate. As Internet access becomes more ubiquitous, demands for mobility are not restricted to single terminals anymore. Mobility is also needed to support the movement of a complete network that changes its point of attachment to the fixed infrastructure, maintaining the sessions of every device of the network—what is known as network mobility in IP networks. In this scenario, the mobile network has at least a (mobile) router that connects to the infrastructure. The devices of the mobile network connect to the exterior through this mobile router.

In the figure, a Mobile IPv6-enabled cellular phone acts as a mobile router.

Summary

This topic summarizes the pimary points that were discussed in this lesson.



Resources

To learn more about IP Mobility, refer to the following material:

- **RFC 3775: Mobility Support in IPv6**
- **IPv6 Extension Headers Review and Considerations** http://www.cisco.com/en/US/technologies/tk648/tk872/technologies white paper0900aecd 8054d37d.html
- Cisco IOS IPv6 Configuration Guide, Release 12.4: **Implementing Mobile IPv6** http://www.cisco.com/en/US/partner/docs/ios/ipv6/configuration/guide/ip6-mobile.html#wp1290551
- **RFC 3963, Network Mobility (NEMO) Basic Support Protocol**
- **RFC 5555, Mobile IPv6 Support for Dual Stack Hosts and Routers**

Lesson 2

Describing DNS in an IPv6 Environment

Overview

Successfully implementing Domain Name System (DNS) is critical to creating a stable and robust IPv6 network. Therefore, it is essential to understand how DNS works, including supported objects, tree structure, and dynamic DNS (DDNS).

DNS is a distributed Internet directory service that is used to translate between domain names and IP addresses, between IP addresses and domain names, to control Internet email delivery, and more. Successfully implementing DNS on IP version 6 (IPv6) is critical because most Internet services rely on DNS to work, and if DNS fails, websites cannot be located and email delivery stalls.

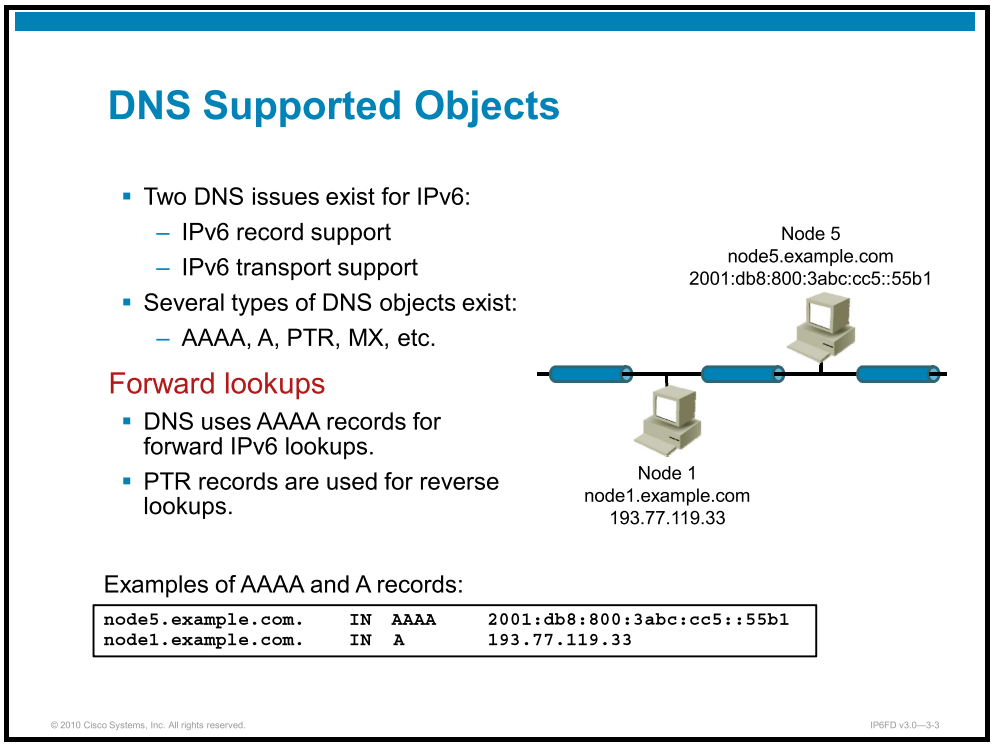
Objectives

Upon completing this topic, you will be able to describe how DNS works in an IPv6 environment. This knowledge includes being able to meet these objectives:

- Identify DNS-supported objects and records in IPv6 networks
- Describe DNS tree structure in IPv6 networks
- Describe how DDNS works in IPv6 networks

DNS Objects and Records

This topic describes DNS-supported objects and records in IPv6 networks.



DNS Basics

The DNS protocol had to be updated to support IPv6 in addition to IPv4. The two main tasks were:

- To enable name lookup for IPv6 addresses
- To enable the servers to communicate between themselves on IPv6 in addition to IPv4

The DNS servers maintain a database for holding the relations between domain names (such as http://www.example.com) and IP addresses. This information is stored in DNS databases in the form of *records*. Depending on the record type (quad-A [AAAA], A, MX, and so on), different information is stored. An MX record, for example, stores the IP address of the mail server for that domain (for example, http://mail.example.com).

Two types of lookups are used most in DNS: *forward* and *reverse*

- Forward lookups provide resolution from a domain name to an IP or IPv6 address.
- Reverse lookups provide resolution from an IP address to a domain name.

Record Types

There are several types of objects in a DNS record about a domain. These include several types of records, such as:

- **A records:** for IPv4 name-to-address lookups
- **AAAA records:** for IPv6 name-to-address lookups
- **MX records:** for the IP address of the mail server

To support IPv6 in DNS, make these two updates to the DNS client and server systems:

- Update the DNS server and client to accept IPv6 record formats.
- Update the DNS server and client to run over both IPv6 and IPv4 transport.

These updates do not have to happen at the same time. Early DNS implementations often support the new AAAA records, but run only over IPv4 transport. These early systems will work only for dual-stack clients and servers. An IPv6-only implementation would not work because DNS would not use IPv6 transport.

Here are the three records or formats for IPv6:

- Forward lookups
- Nibble format (reverse lookups)
- Bitwise format (reverse lookups)—deprecated

Bitwise format is no longer recommended and has been moved to experimental status, but some implementations still deploy it.

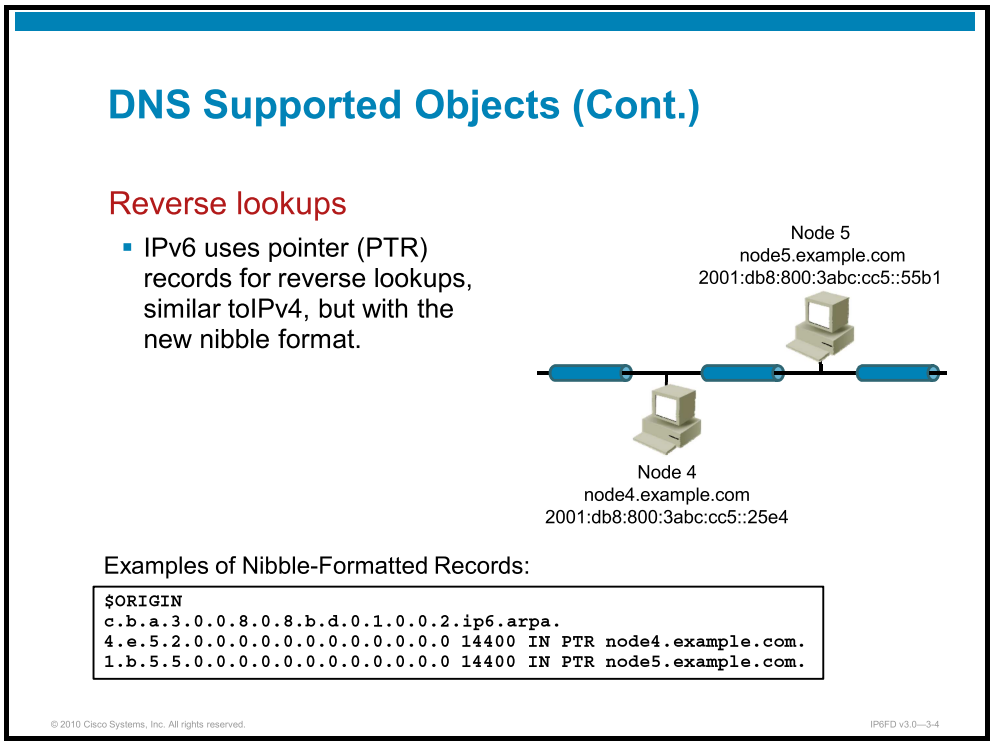
Forward Lookups

Forward lookups (name to address) are completed via the AAAA record (quad A), which is the address record for IPv6 DNS. This record links a hostname to a 128-bit address, which is the forward lookup record.

Here is an example of a AAAA record:

```
$ORIGIN example.com.
node4 3600 IN AAAA 2001:db8:800:3abc:cc5::25e4
node5 3600 IN AAAA 2001:db8:800:3abc:cc5::55b1
```

Note There were A6 records to resolve an IPv6 address from a name, however, they are deprecated. AAAA records are used instead.



Reverse Lookups

Reverse lookups (address to name) are still accomplished using the pointer (PTR) record. There are two formats for address representation: one recommended and one deprecated, which is *not* recommended.

The nibble format is preferred. It uses the top-level domain “ip6.arpa” (initially the top-level domain was called “ip6.int,” but that convention was deprecated in RFC 4159 and need not be maintained any longer). Notice that in the following example, that address representation is backward, with each 4-bit position (one hexadecimal character) separated by a “.” (dot). There is no compressed format for the address, so you cannot eliminate leading zeros.

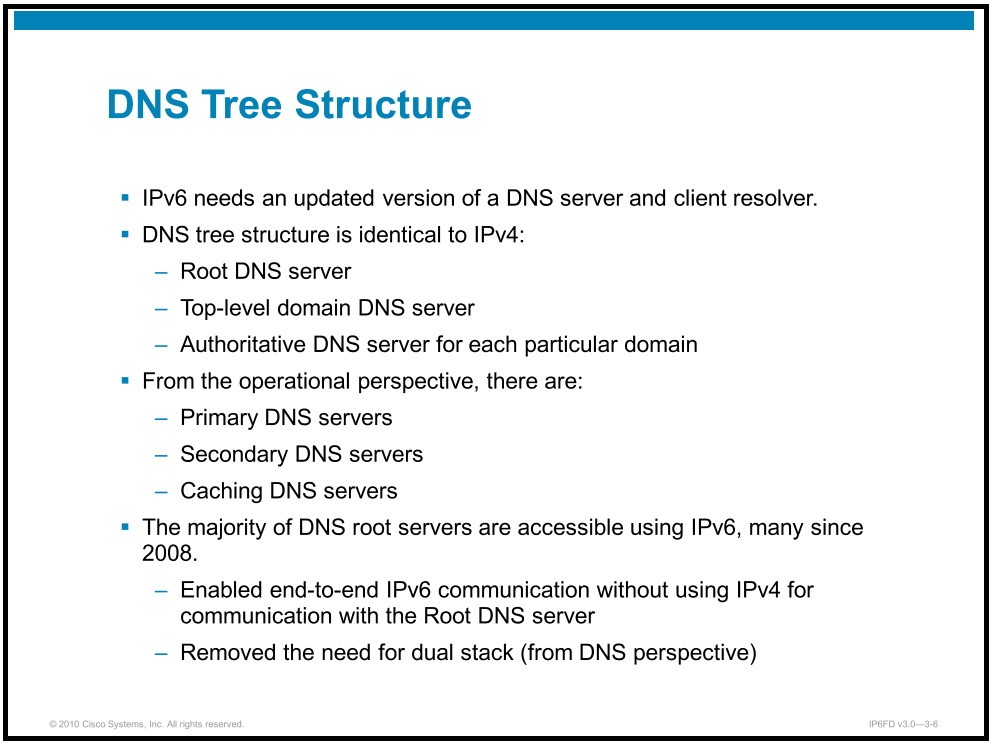
```
$ORIGIN c.b.a.3.0.0.8.0.8.b.d.0.1.0.0.2.ip6.arpa.
4.e.5.2.0.0.8.0.8.b.d.0.1.0.0.2 14400 IN PTR node4.example.com.
1.b.5.5.0.0.8.0.8.b.d.0.1.0.0.2 14400 IN PTR node5.example.com.
```

The bitwise, or “bitlabel,” format is no longer preferred. The format was specified in RFC 2673, *Binary Labels in the Domain Name System,* in August 1999 as a proposed standard, but was moved to experimental status by RFC 3363, *Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS),* in August 2002. Apparently, some DNS implementations were rejecting the bitlabel format queries as “malformed,” rather than returning a PTR record (where one existed) or “none found” (which would allow the querier to switch to nibble format and proceed). Notice, in the following example, that this representation is forward-specified.

```
$ORIGIN \[x20010db808003abc/64].ip6.arpa
\[x00000000000025e4/64] 14000 IN PTR node4.example.com
\[x00000000000055b1/64] 14000 IN PTR node5.example.com
```


DNS Tree Structure

This topic describes DNS tree structure in IPv6 networks.



DNS Server Hierarchy

The hierarchy of DNS servers is best described with a tree. On the top of the hierarchy, there are root DNS servers (only 13 clusters of these servers in the world).

Below the root server are top-level domain DNS servers (TLDs), which resolve IP addresses for top-level domains, such as .com, .net, .org, .us, .uk, and so on.

Beneath TLD servers, there are Authoritative servers for each domain. These resolve IP addresses from their domains only (such as, for http://example.com).

DNS Server Tree Structure

The IPv6 DNS tree structure is identical to the deployed structure for IPv4. Clients query local caching servers, which locate the DNS server with the authoritative records for a given zone through message exchange with a root DNS server. They then return records to the client (and cache the information locally for near-term future use). Typically, a protocol-independent application will query for both an A and an AAAA record, and then generally prefer the IPv6 path.

These major components of the DNS tree structure are included:

- Root DNS
- Primary authoritative DNS
- Secondary authoritative DNS
- Caching DNS (typically also deployed in sets, not a single machine)
- Client-based DNS resolver library

For redundancy and operational efficiency reasons, there are primary and secondary DNS servers for every hierarchy level, and cache DNS servers that cache results of DNS queries within enterprise networks.

Root DNS Servers

Root DNS servers contain records that link domain names to their authoritative DNS servers.

The records that are maintained by the root DNS server for a given domain name should include at least one IPv4 address for the authoritative server (for a given zone). The records may contain more than one IPv4address and may also contain multiple IPv6 addresses. This inclusion prevents situations in which an IPv4-only caching server is referred to as an IPv6-only authoritative server, to which it could clearly not connect.

An example record (in generalized format) would be:

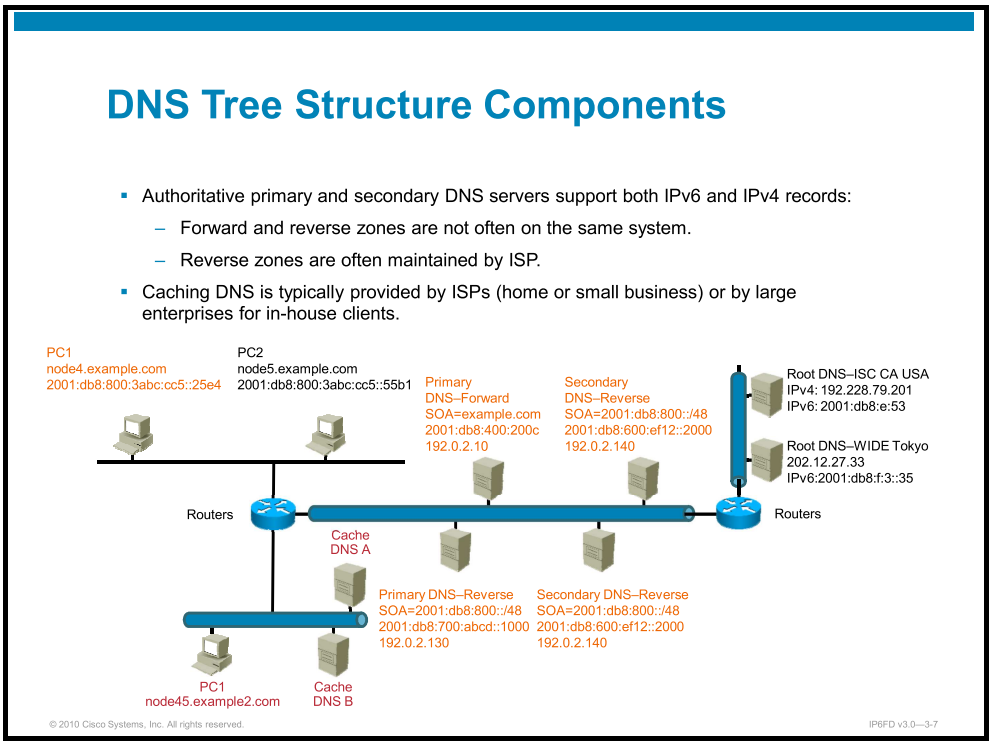
```
"example.com – Authoritative Primary DNS at 2001:db8:400::200c,
Secondary at 2001:db8:100::4e20"
"example.com – Authoritative Primary DNS at 192.0.2.10, Secondary at
192.0.2.20"
"2001:db8:800:3abc:cc5::25e4" – Authoritative Primary
(2001:db8:800/48) at 2001.db8:700:abcd::1000,
Secondary at 2001:db8:600:ef12::2000
"2001:db8:800:3abc:cc5::25e4" – Authoritative Primary
(2001:db8:800/48) at 192.0.2.130,
Secondary at 192.0.2.140
```

The fact that the root DNS servers only advertise their IPv4 addresses (even though some do respond on IPv6) means that it was not possible to deploy an IPv6-only enterprise. Within the enterprise, all DNS servers can be implemented using IPv6 transport: "Resolver client to local caching server."

The problem is that the caching server had to talk to the root DNS over IPv4 transport, so it had to be a dual-stack node, not an IPv6-only node.

Furthermore, many of the country code top-level domain (ccTLD) DNS servers (servers providing a domain name for a specific country code, such as ".us") were also IPv4-only.

Note IANA added AAAA records for its root DNS servers in 2008. Since then, approximately half of the servers are reachable using IPv6, making IPv6-only networks possible.



DNS Tree Structure Components

Root DNS

Root DNS servers contain records that link domain names to their authoritative DNS servers. There are currently 13 root DNS IP addresses (there are more than 13 servers—many addresses are IPv4-anycast addresses and “front” a number of servers). The root DNS servers are not uniformly addressable on their IPv6 addresses; some are reachable over IPv6 transport, but several are still not.

Top-Level DNS

These servers resolve IP addresses for TLDs, such as .com, .net, .org, .info, .biz, and for country TLDs (ccTLDs), such as .us, .uk, .de, .hk, .au, and so on.

Authoritative Primary DNS

For a given domain, authoritative primary DNS servers contain the official records for hosts within a given domain name. For reverse lookups, authoritative primary DNS servers contain the official reverse-lookup records for the given IP address. Typically, the forward authoritative DNS server is not the same host as the secondary authoritative DNS server.

Examples of records that are maintained on these DNS servers are:

```
“node4.example.com” - 2001:db8:800:3abc:cc5::25e4
“2001:db8:800:3abc:cc5::25e4” - node4.example.com
```

Secondary DNS

For a given domain, secondary DNS servers provide a backup in case the primary DNS server fails. Secondary DNS servers periodically transfer records from the primary DNS server.

Caching DNS

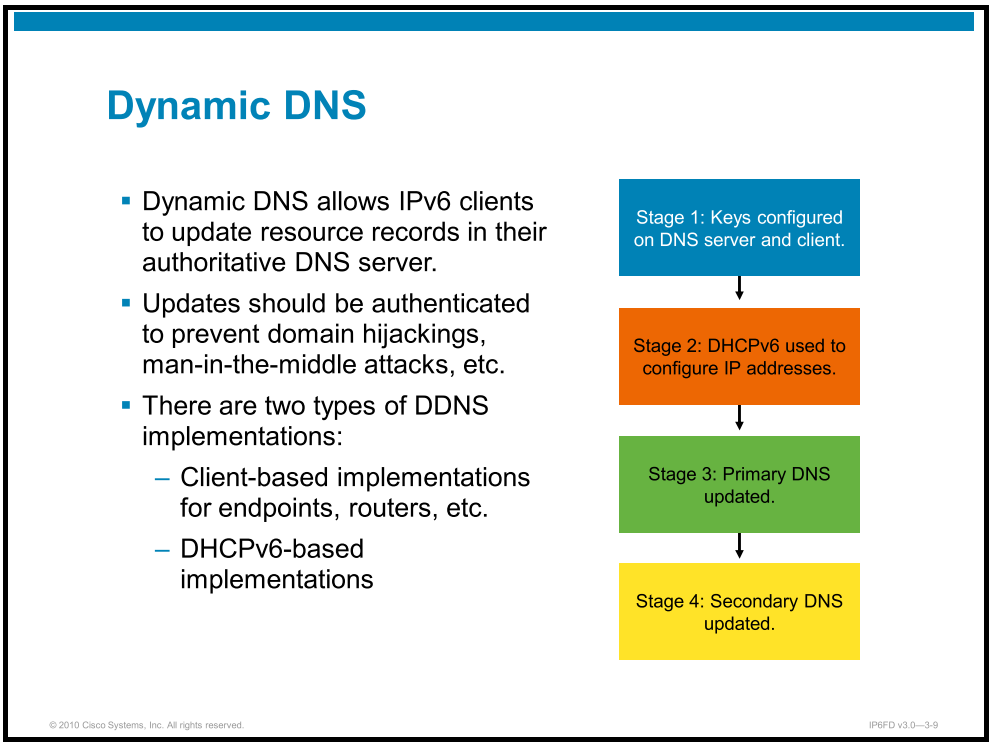
Caching DNS servers answer queries from client devices and help reduce the load on the primary, secondary, and root DNS servers. No records are permanently maintained on caching DNS servers. When a caching DNS server helps resolve a record on behalf of a client, it stores the record locally in a cache for a time—to use when answering other clients asking for the same record—before discarding it.

Client Devices

Client devices are IP nodes that use a DNS resolver to translate names to addresses and addresses to names. Client devices are configured to point to multiple caching servers.

Dynamic DNS

This topic describes how DDNS works in IPv6 networks.



Historically, IP-based servers had their addresses manually configured into the primary DNS server. These addresses were also statically assigned on the node; therefore, the addresses and name-to-address translation was long-lived. For example, hostname “media.example.com” would be at IP address 192.168.0.200, and that entry would be in the primary (and secondary) authoritative DNS server. Client machines usually did not have an entry in DNS, because the ability to be reliably contacted by a peer node was not desired or practical.

When most devices are both clients and servers (in other words, peers—an important driver for IPv6 adoption) and those devices configure their addresses via, for example, DHCPv6 or autoconfiguration, those nodes need to dynamically create or update their DNS records on their authoritative primary DNS server. That gives them stable host-to-address and address-to-host mappings even when their dynamically assigned addresses change. Using DDNS, DHCPv6 clients can dynamically update their records in DNS. DDNS is still under active discussion in the Internet Engineering Task Force (IETF) working groups. Many published RFCs and drafts that are related to DDNS are in circulation.

The DDNS process goes through these stages:

- **Stage 1:** Keys are configured on the DNS server and client. DDNS exchanges must be secured, otherwise man-in-the-middle attacks, in which a malicious party captures traffic intended for another node, are possible.
- **Stage 2:** IPv6 node uses DHCPv6 to configure an IP address or other information. The address can also be configured via stateless autoconfiguration, and the DDNS update can be performed in the same manner.
- **Stage 3:** The primary DNS is updated. The DHCPv6 client on the node updates the primary DNS server for both forward and reverse records.

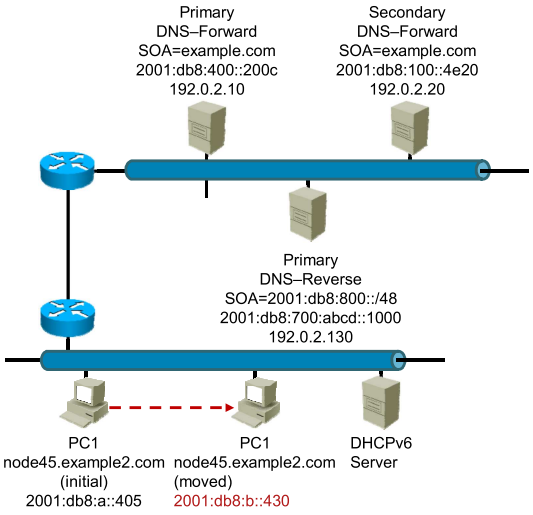
- **Stage 4:** The Secondary DNS is updated. The primary DNS server updates the secondary DNS server via zone transfer.

Tip DDNS is an important building block for IPv6, even though it is not strictly related to IPv6. In IPv4, most network applications are client-server, such as web servers and browsers. In IPv4 application architecture, clients are normally anonymous—they have no entry in the global DNS system—and servers are only reachable at well-known DNS names.
One compelling IPv6 feature is the ability to support peer computing, in which the terms “client” and “server” are no longer meaningful. All nodes are complete peers on the network and reachable via their well-known DNS name. This scenario implies that all nodes have current entries in the DNS. From a scalability perspective, and considering that many nodes will be mobile and will use autoconfiguration or DHCPv6 for address assignment, DDNS is the only reasonable solution to ensure that these nodes always have a current DNS mapping.

Dynamic DNS Process

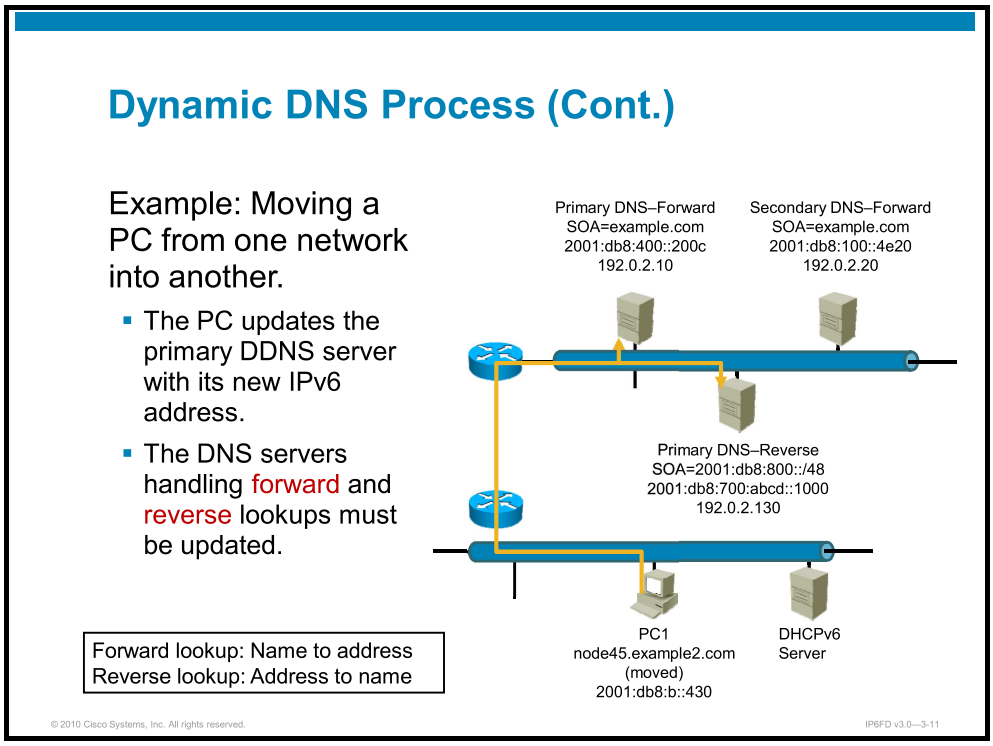
Example: Moving a PC from one network into another.

- The PC must maintain a relationship with a DDNS server.
- It needs to be reachable using a DNS name.



The figures show an example in which a PC is physically moved from one network to another. When the PC is turned back on, it exchanges traffic with the DHCPv6 server and receives a new IP address.

Once the PC has been given the new IP address, it can exchange packets with the two DNS servers in the network.



The Primary DNS–Forward server holds the authoritative name-to-address records for the zone. The Primary DNS–Reverse server holds the authoritative address-to-name records for the address range. As a result, the two primary DNS servers are updated, and the secondary servers learn the modified records from the primary servers.

Forward Resolution and Reverse Registration

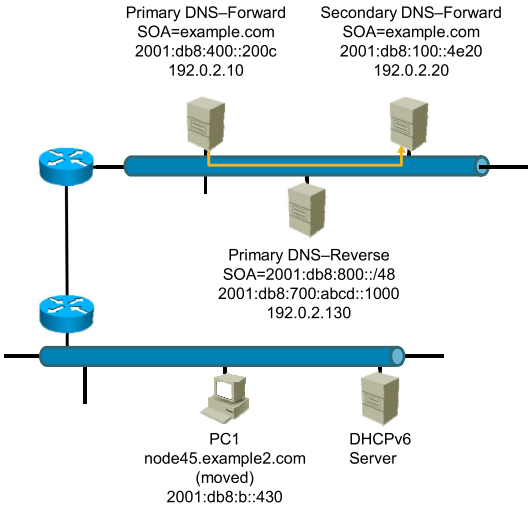
DDNS works well for forward resolution, because the node will always register with the same authoritative forward DNS server (the server for the given domain name). For example, bob.example.net will always register with the example.net authoritative forward DNS server. Here, the same organization is likely to own and operate both nodes, the node changing addresses and the forward DNS server, and a trust relationship can normally be established.

DDNS works less well for reverse registration, because the node may have moved to a completely foreign network (for example, coffeeshop.com). In this case, the node would not have exchanged keys with the authoritative reverse DNS server and would not be able to update the reverse zone. Here, the same organization does not manage the node and the DNS server. No prearranged trust relationship is likely to exist, so DDNS updates would fail.

Dynamic DNS Process (Cont.)

Example: Moving a PC from one network into another.

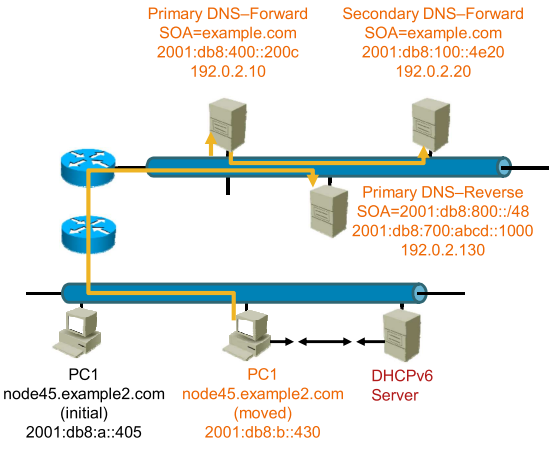
- The secondary DNS server is updated.
- The PC is reachable from the Internet to its new, updated IPv6 (or IPv4) address.



Finally, the secondary DNS server is updated using zone transfer functionality. DNS zone transfer is one of the ways that DNS servers replicate their databases.

Dynamic DNS and DHCPv6

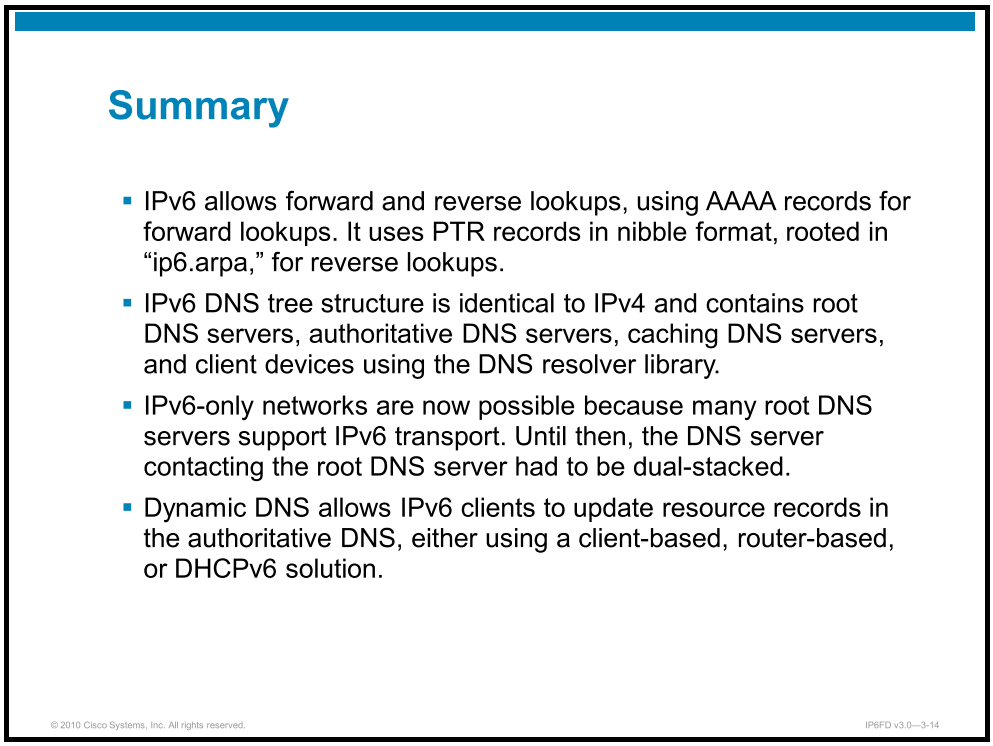
- The DHCPv6 client address may be updated in DNS by the DHCPv6 server.
- It is more often used when DHCP and DNS are tightly integrated.
- No integrated DHCPv6 or DNS is available.



In some implementations, DHCP and DNS are functions, and the DHCP server dynamically updates DNS on behalf of the client node. This functionality effectively moves the trust requirement from between the node and the DNS server to being, instead, between the DHCPv6 server and the DNS server.

Summary

This topic summarizes the primary points that were discussed in this lesson.



Resources

To learn more about DNS, refer to the following resources:

- *IANA: IPv6 Addresses for the Root Servers* at http://www.iana.org/reports/2008/root-aaaa-announcement.html
- *List of DNS Record Types* at http://en.wikipedia.org/wiki/List_of_DNS_record_types
- *Root nameserver* at http://en.wikipedia.org/wiki/Root_nameserver

Lesson 3

Understanding DHCPv6 Operations

Overview

One highly touted benefit of IP version 6 (IPv6) is its autoconfiguration capability. At first glance, it might appear that autoconfiguration would lessen or even alleviate the requirements for a process like DHCP. However, in some managed environments, network administrators will want to control who accesses network resources and manage address allocation. Consequently, DHCP will continue to be a valuable service in modern networks. This lesson describes DHCP version 6 (DHCPv6) for IPv6 operations, including how DHCP operation in IPv6 differs from its operation in IP version 4 (IPv4) and how you can implement DHCPv6 prefix delegation to improve the IPv6 numbering process.

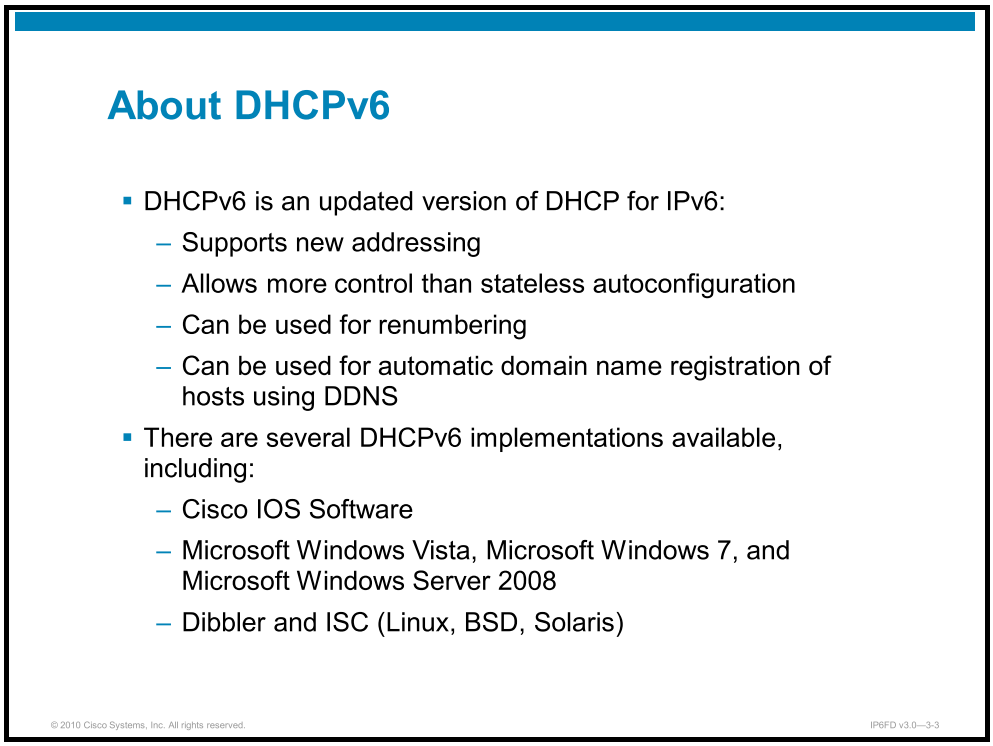
Objectives

Upon completing this lesson, you will be able to describe how DHCPv6 operates. This ability includes being able to meet these objectives:

- Describe how DHCP operation in IPv6 differs from its operation in IPv4
- Describe the operation of DHCP in IPv6
- Identify the multicast addresses that DHCP uses in IPv6
- Describe how DHCPv6 prefix delegation works
- Troubleshoot DHCPv6

DHCPv6

This topic describes th features of DHCPv6.

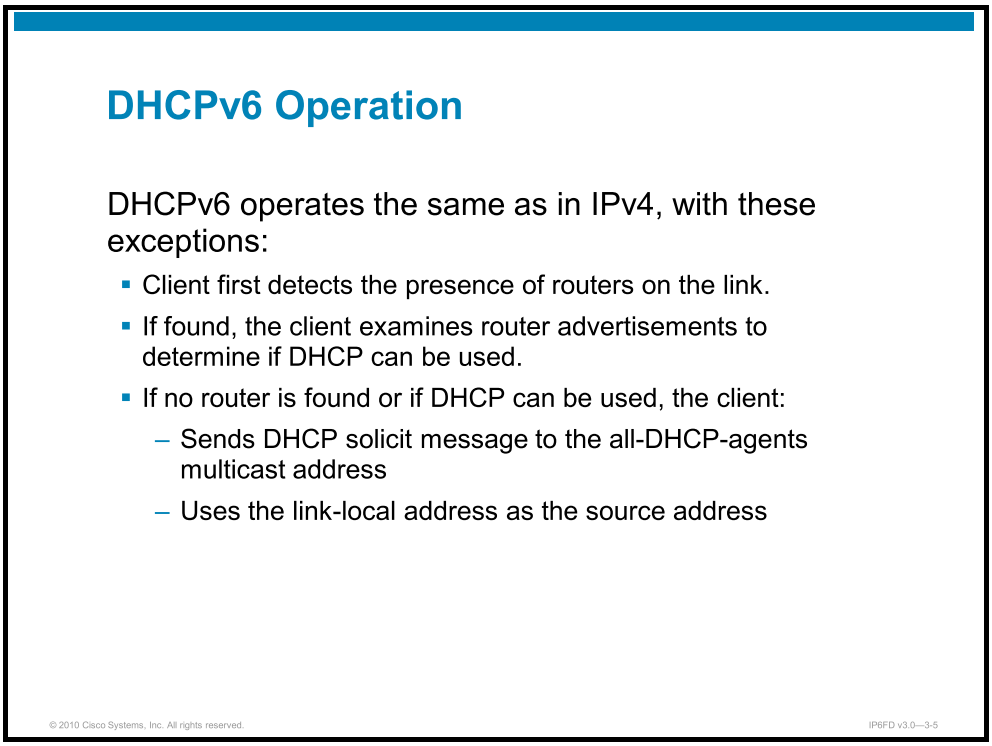


DHCPv6 is an updated version of DHCP for use with IPv6. It supports the addressing model of IPv6 and benefits from new IPv6 features. Some of these features include:

- DHCPv6 enables more control than serverless or stateless autoconfiguration.
- It can function in a routerless environment, using only servers.
- It can be used concurrently with stateless autoconfiguration.
- DHCPv6 can be used for renumbering.
- It can be used for automatic domain name registration of hosts using the Dynamic Domain Name System (DDNS).
- DHCPv6 was ratified in RFC 3315 (July 2003).

DHCPv6 Operation

This topic describes how DHCPv6 operates.



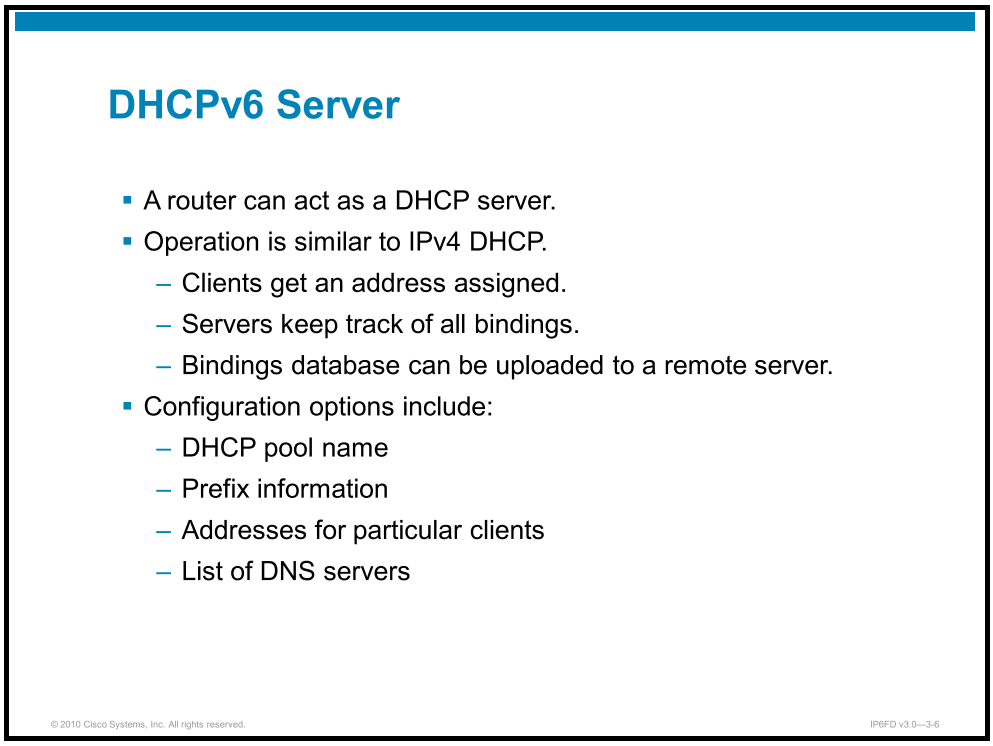
Acquiring configuration data for a client in DHCPv6 is like the process in IPv4 but with a few exceptions. The client can sometimes detect the presence of routers on the link using neighbor discovery messages. If at least one router is found, the client examines the router advertisements to determine if DHCP should be used. If the router advertisements allow use of DHCP on that link or if no router is found, the client starts a DHCP solicit phase to find a DHCP server.

DHCPv6 uses multicast for many messages. When the client sends a solicit message, it sends the message to the all-DHCP-agents multicast address with link-local scope. Agents include both servers and relays.

When a DHCP relay forwards a message, it can forward it to the all-DHCP-servers multicast address with site-local scope. This means that a relay does not need to be configured with all the static addresses of the DHCP servers, as in IPv4. If needed by policy, a relay can contain a static list of DHCP servers.

Some servers can be configured to give global addresses using policies, for example, “do not give to printer.” Other servers (or the same servers within a different context) can be configured to give site-local addresses using a different policy, for example, “give to anyone.”

Tip DHCPv6 solicit messages are sent from the link-local address of the requesting node—the address that the node constructs for itself at initialization. The request is sent to a reserved DHCPv6-specific multicast address. This process differs markedly from the IPv4 practice, in which the message is sent from the unspecified address to the broadcast address. This scenario is an excellent example of IPv6 using more elegant mechanisms than IPv4 to improve network scalability.



The DHCPv6 server function can be enabled on individual IPv6-enabled interfaces.

The DHCPv6 server can provide those configuration parameters that do not require the server to maintain any dynamic state for individual clients, such as DNS server addresses and domain search list options. The DHCPv6 server may be configured to perform prefix delegation.

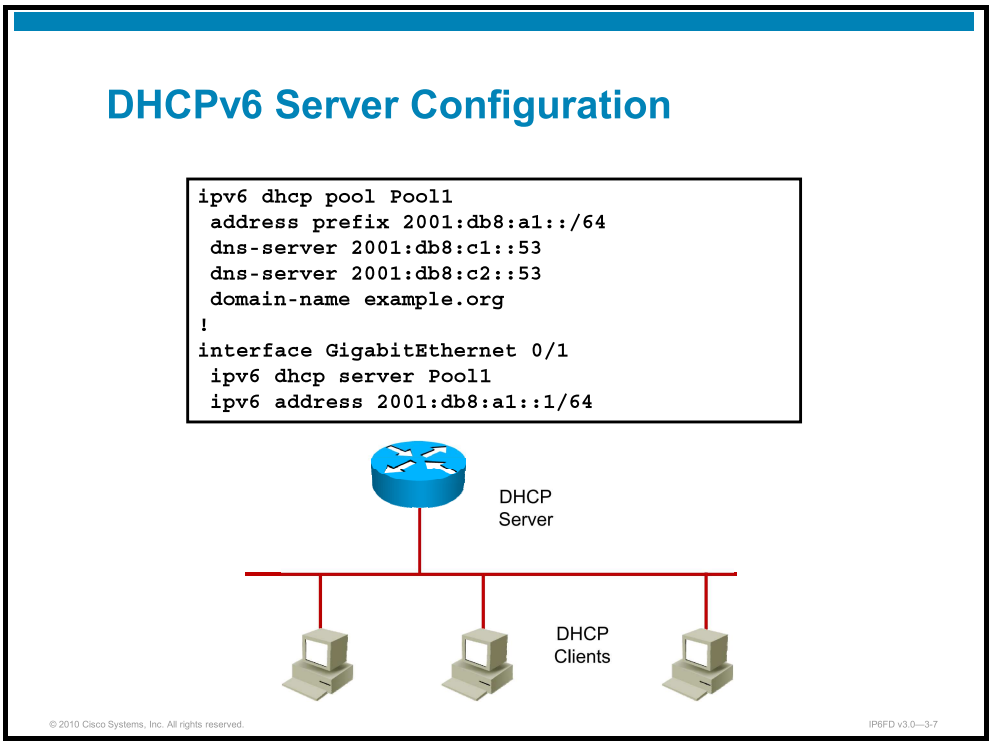
All the configuration parameters for clients are independently configured into DHCPv6 configuration pools, which are stored in NVRAM. A configuration pool can be associated with a particular DHCPv6 server on an interface when it is started. Prefixes to be delegated to clients may be specified either as a list of preassigned prefixes for a particular client or as IPv6 local prefix pools, which are also stored in NVRAM. DHCPv6 configuration pools can reference and use the list of manually configured prefixes or IPv6 local prefix 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.

A DHCPv6 configuration information pool is a named entity that includes information about available configuration parameters and policies that control assignment of the parameters to clients from the pool. A pool is configured independently of the DHCPv6 service and is associated with the DHCPv6 service through the command-line interface (CLI).

Each configuration pool can contain the following configuration parameters and operational information:

- Prefix delegation information, which can include:
  - A prefix pool name and associated preferred and valid lifetimes
  - A list of available prefixes for a particular client and associated preferred and valid lifetimes
- A list of IPv6 addresses of DNS servers.
- A domain search list, which is a string containing domain names for DNS resolution.



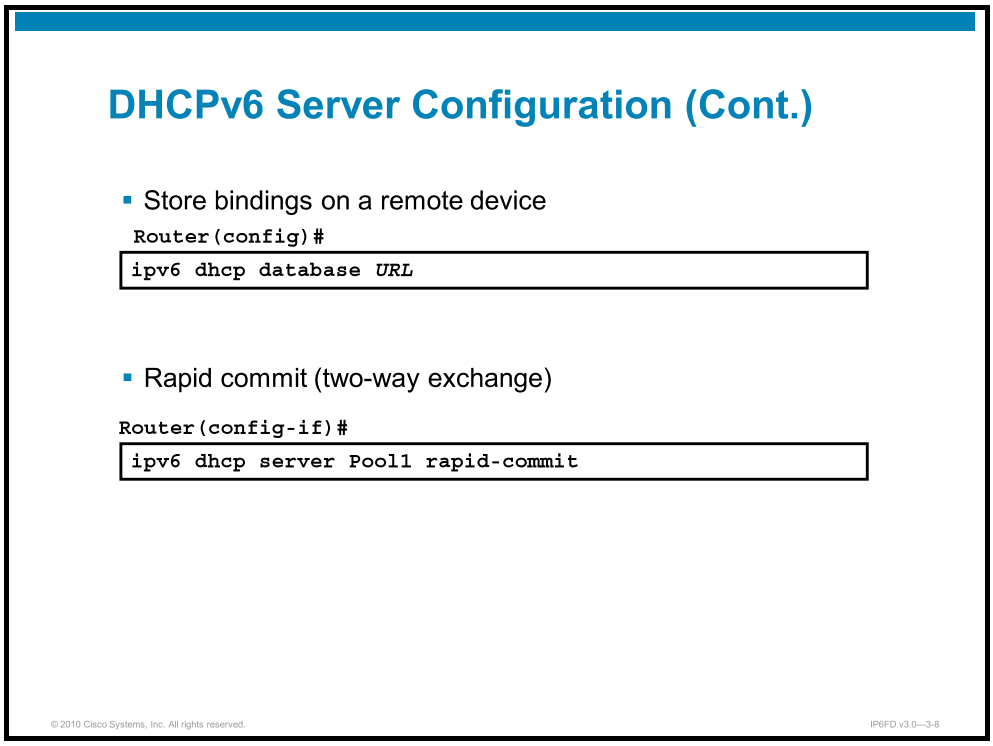
The figure provides an example of DHCP server configuration on Cisco IOS Software.

```
ipv6 dhcp pool Pool1
 address prefix 2001:db8:a1::/64
 dns-server 2001:db8:c1::53
 dns-server 2001:db8:c2::53
 domain-name example.org
!
interface GigabitEthernet 0/1
 ipv6 dhcp server Pool1
 ipv6 address 2001:db8:a1::1/64
```

There is a DHCP pool named "Pool1," which has the following configuration:

- **Address prefix:** 2001:db8:a11::/64
- **DNS server:** 2001:db8:c1::53
- **DNS server:** 2001:db8:c2::53
- **Domain name:** example.org

On the interface GigabitEthernet 0/1, use the **ipv6 dhcp server** command to activate this pool.



Other configuration options are:

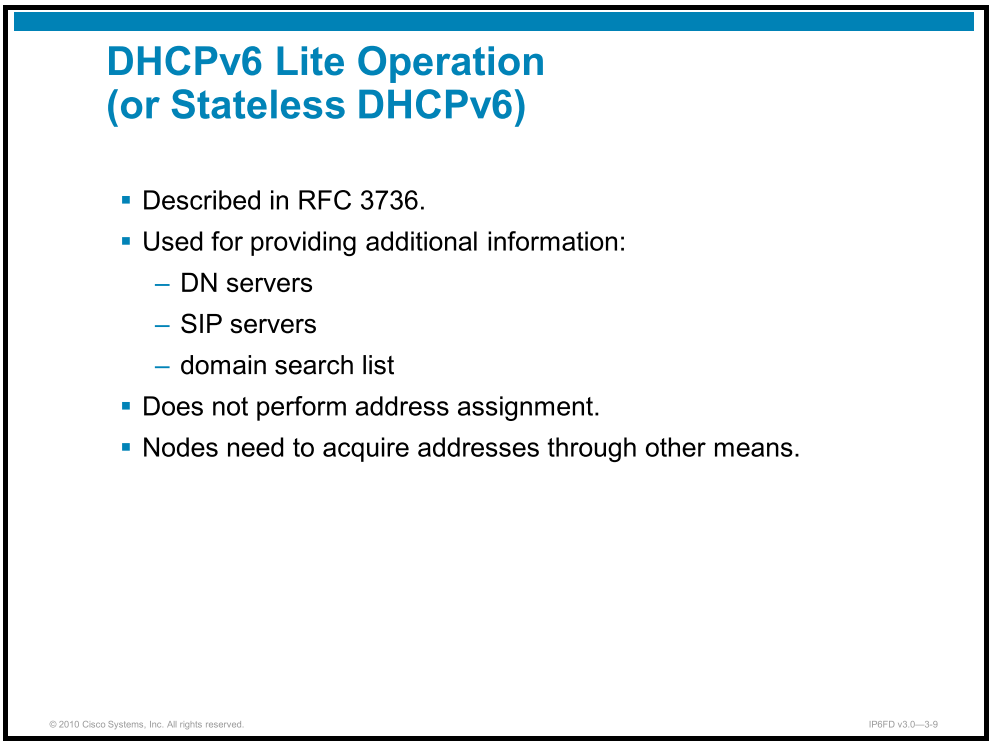
- Remote DHCP binding database
- Rapid commit

A binding table entry is automatically created whenever a prefix is delegated to a client from the configuration pool. The binding table entry is updated when the client renews, rebinds, or confirms the prefix delegation. It is deleted when the client releases all the prefixes in the binding voluntarily, the valid lifetimes of all prefixes have expired, or administrators enable the **clear ipv6 dhcp binding** command.

These bindings are maintained in RAM and can be saved to permanent storage using the *agent* argument. In this way, the information about configuration—such as prefixes assigned to clients—is not lost after a system reload or power-down. The bindings are stored as text records for easy maintenance.

Each permanent storage facility, to which the binding database is saved, is called the database agent. A database agent can be a remote host such as an FTP server or a local file system such as NVRAM.

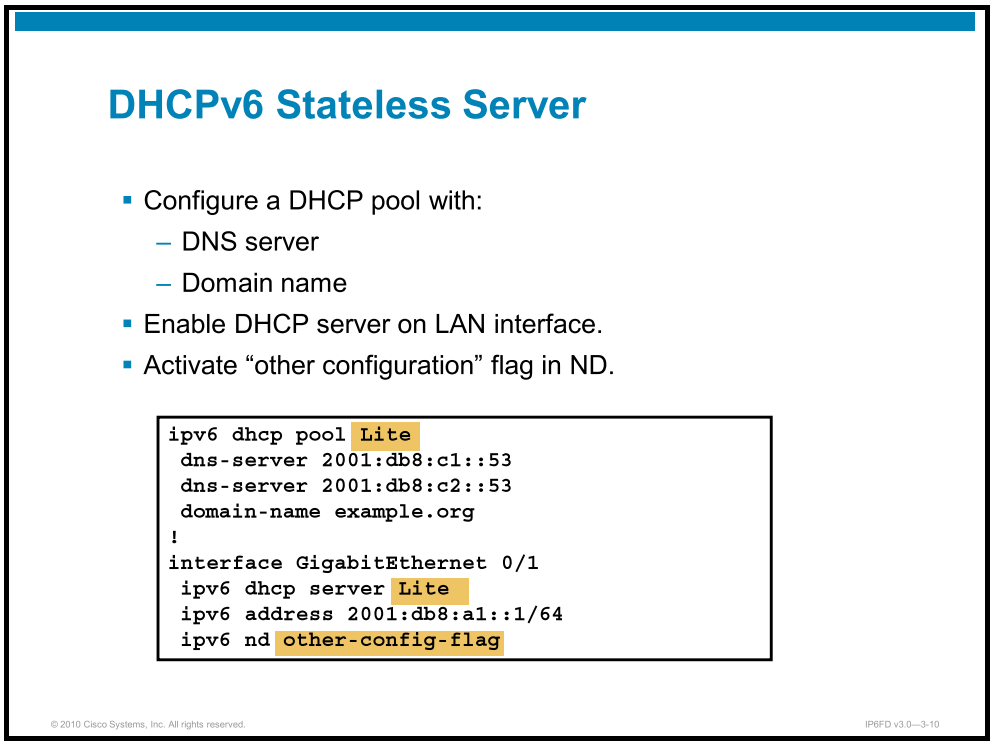
The DHCPv6 client can obtain configuration parameters from a server either through a rapid two-message exchange (solicit, reply) or through a normal four-message exchange (solicit, advertise, request, reply). By default, the four-message exchange is used. When both the client and server enable the rapid-commit option, the two-message exchange is used.



DHCPv6 Lite (or stateless DHCPv6) is used in an environment where end nodes acquire IPv6 addresses through different means (most often using stateless autoconfiguration). However, they also need to obtain additional information (usually, a list of DNS servers).

A stateless DHCPv6 server will send additional information if contacted by a stateless client. The client will be updated with additional information through DHCP Lite by using a router advertisement (RA) message. The router needs the command **ipv6 nd ...** to be configured on the interface that advertises router presence

A Cisco router running Cisco IOS Software can also act as a stateless DHCPv6 client if so instructed by another router with RA messages.



The figure describes how to use the DHCPv6 function to configure clients with information about the name lookup system. The server does not maintain a state that is related to clients; for example, prefix pools and records of allocation are not maintained. Therefore, this function is "stateless" DHCPv6.

Configuration Options

Command	Description
ipv6 dhcp pool ***poolname***	Configures a DHCPv6 configuration information pool and enters DHCPv6 pool configuration mode
dns-server ***ipv6-address***	Specifies the DNS IPv6 servers available to a DHCPv6 client
domain-name ***domain***	Configures a domain name for a DHCPv6 client
ipv6 dhcp server ***poolname***	Enables DHCPv6 on an interface
ipv6 nd other-config-flag	Sets the "other configuration" flag in IPv6 RA messages

Note Cisco IOS Software can act as a DHCP client on any router interface. To achieve this, the command **ipv6 address dhcp** needs to be configured on that interface. If rapid commit is needed, the optional setting **rapid-commit** needs to be configured on both the client and the server.

A DHCP relay agent, which may reside on the client's link, is used to relay messages between the client and the server. DHCP relay agent operations are transparent to the client. A client locates a DHCP server using a reserved, link-scoped multicast address. Therefore, direct communication between the client and the server requires that the client and the server are attached to the same link. However, sometimes—when ease of management, economy, or scalability is a concern—allowing a DHCP client to send a message to a DHCP server that is not connected to the same link is desirable.

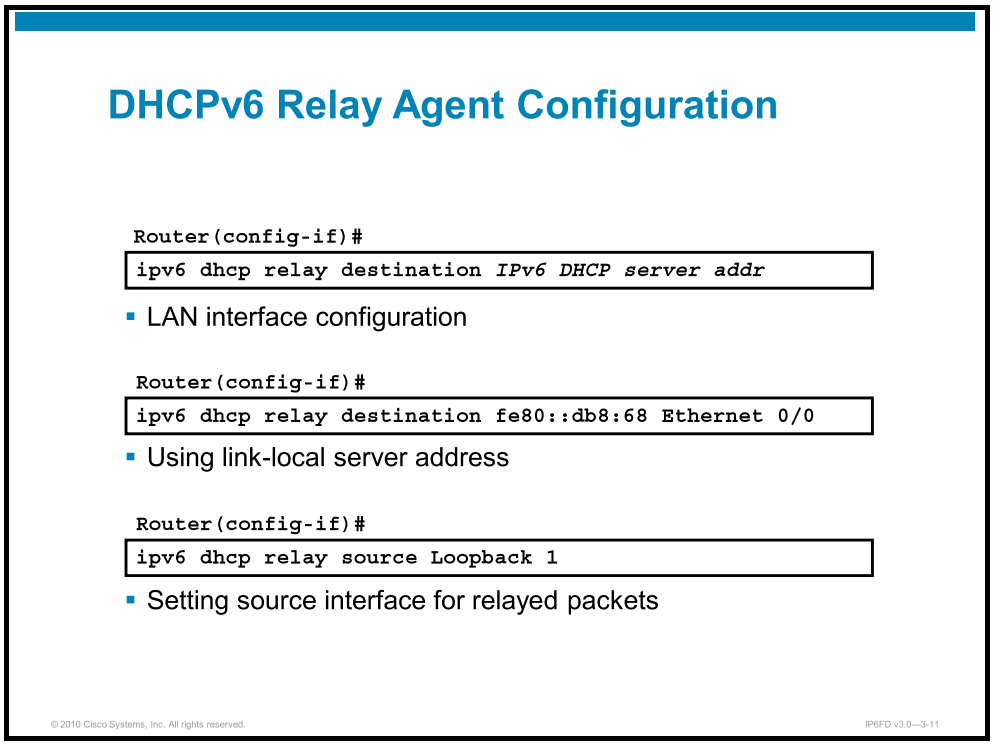
DHCPv6 Relay Agent Notification for Prefix Delegation

DHCPv6 relay agent notification for prefix delegation allows the router working as a DHCPv6 relay agent to find prefix delegation options. These options are found by reviewing the contents of a DHCPv6 RELAY-REPLY packet that the relay agent relays to the client. When the relay agent finds a prefix delegation option, the relay agent extracts the information about the delegated prefix. The relay agent then inserts an IPv6 static route matching the prefix delegation information onto the relay agent. Future packets that are destined to that prefix via relay will be forwarded based on the information contained in the prefix delegation. The IPv6 static route is then left in the routing table until the prefix delegation lease time expires or the relay agent receives a release packet from the client releasing the prefix delegation.

User configuration is not required for this feature. The relay agent completes static route management automatically.

The IPv6 routes are added when the relay agent relays a RELAY-REPLY packet. The IPv6 routes are deleted when the prefix delegation lease time expires, or the relay agent receives a release message. An IPv6 static route in the routing table of the relay agent can be updated when the prefix delegation lease time is extended.

This feature leaves a static IPv6 route on the routing table of the relay agent. This registered IPv6 address allows unicast reverse path forwarding (uRPF) to work by allowing the router doing the reverse-lookup to confirm that the IPv6 address on the relay agent is not malformed or spoofed. The static route that is left in the routing table of the relay agent can be redistributed to other routing protocols to advertise the subnets to other nodes. The static routes will be removed when the client sends a DHCP DECLINE message.

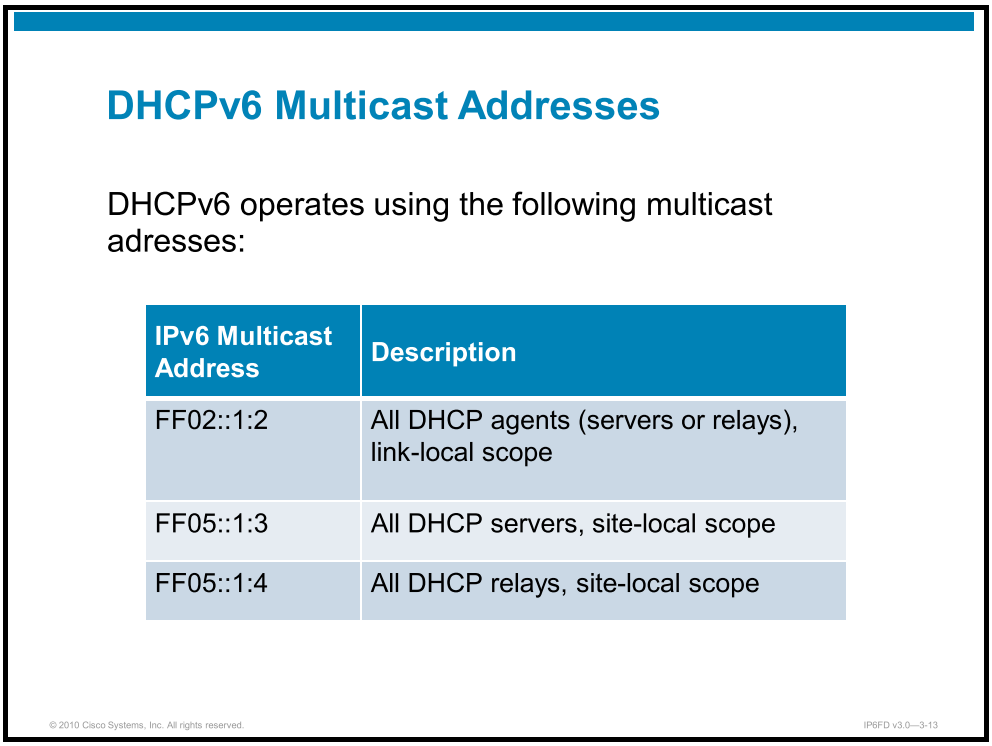


The figure shows the necessary commands to enable the DHCPv6 relay agent function and specify relay destination addresses on an interface. If a link-local address is used for a DHCP server, then the exit interface must also be configured.

The DHCPv6 server sends its replies to the source address of relayed messages. Normally, a DHCPv6 relay uses the address of the server-facing interface that is used to send messages as the source. However, in some networks, it may be desirable to configure a more stable address (such as a loopback interface) and have the relay use that interface as the source address of relayed messages. The DHCPv6 relay source configuration feature provides this capability.

DHCPv6 Multicast Addresses

This topic describes the IPv6 multicast addresses that DHCPv6 uses.



DHCPv6 uses these multicast addresses:

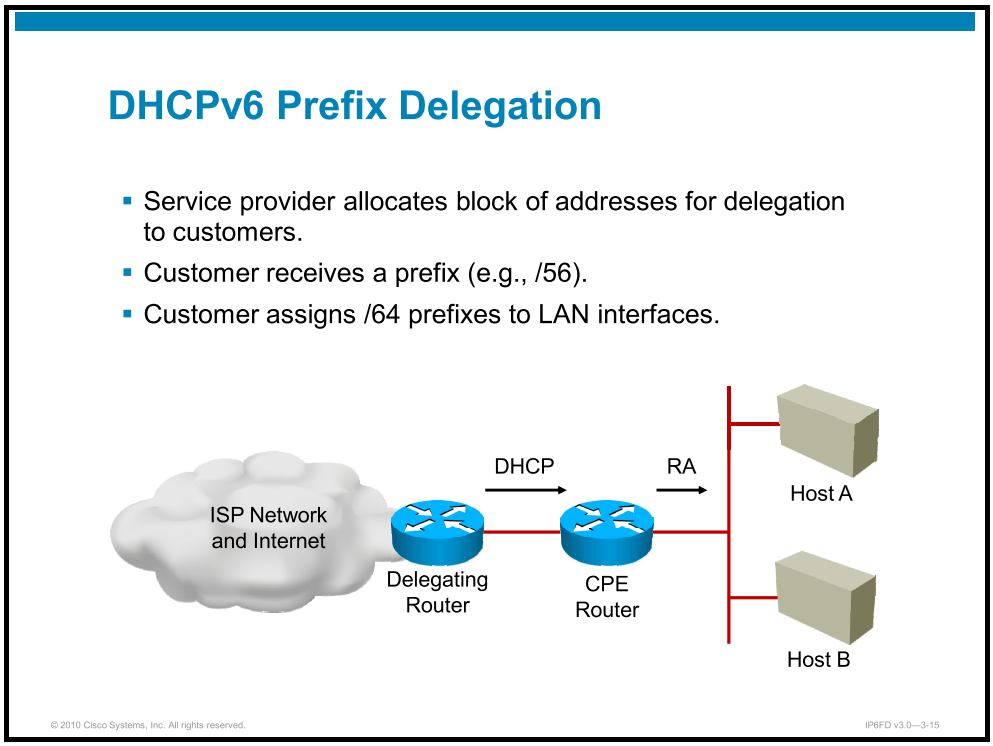
- FF02::1:2 is the all-DHCP-agents (servers or relays) address that clients use to communicate with unknown agents on their local link. This address is of link-local scope.
- FF05::1:3 is the all-DHCP-servers address that relays (or clients) use to communicate with unknown site-wide servers. This address is of site-local scope. A node sending a message to this destination address should use a source address that will be reachable by the server for the answer. For example, a link-local address cannot be used as a source address for this kind of message.
- FF05::1:4 is the all-DHCP-relays address. Because it is an all-DHCP-servers address, it can be used to reach all DHCP relays within one site.

Note that DHCPv6 servers and relay agents listen on UDP port 547, while DHCPv6 clients listen on UDP port 546.

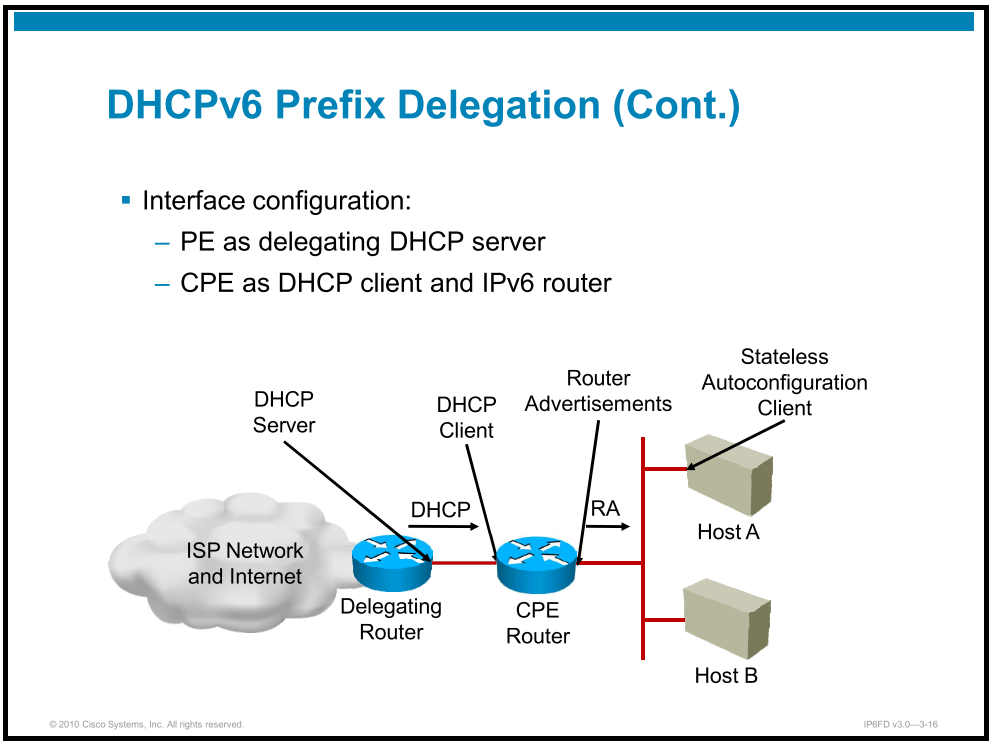
Tip An enterprise would need to have a wide-area (site-wide, at least) multicast implementation to take advantage of the site-scoped DHCPv6 servers address. If the enterprise did not have that enabling mechanism in place, the relays could also be configured with the unicast IPv6 addresses of the servers. Cisco IOS Software requires relays to specifically enumerate the location of DHCPv6 servers.

DHCPv6 Prefix Delegation Process

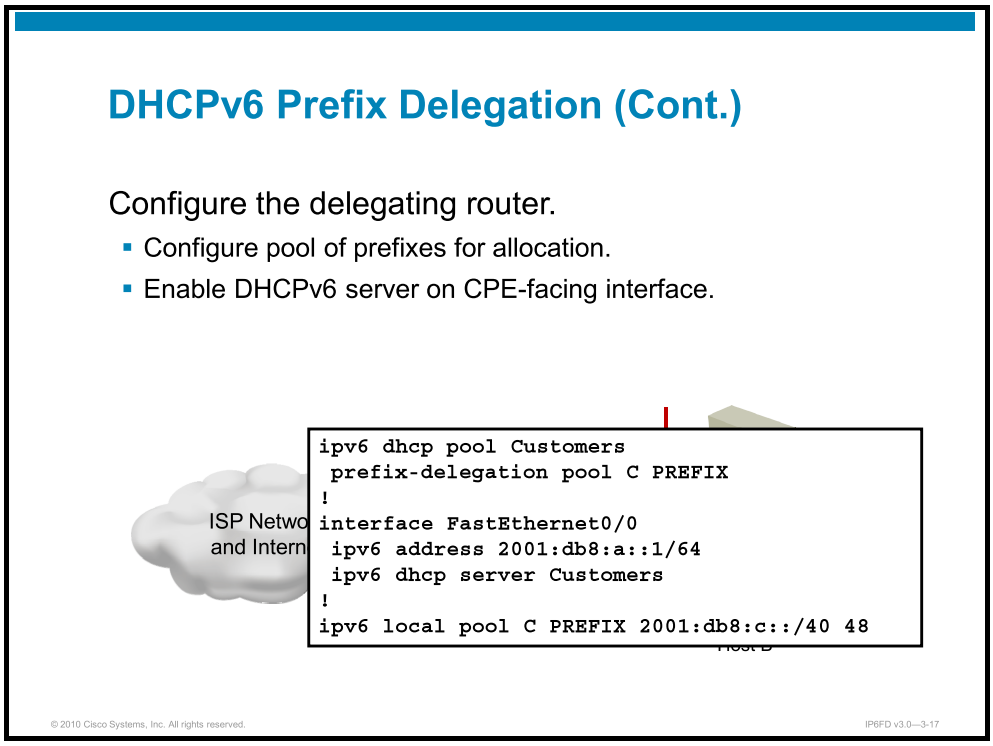
This topic describes some configuration specifics for DHCPv6.



Extensions to DHCPv6 enable prefix delegation, through which an ISP can automate the process of assigning prefixes to a customer for use within the customer network. Prefix delegation occurs between a provider edge (PE) device and customer premises equipment (CPE), using the DHCPv6 prefix delegation option. After the ISP has delegated prefixes to a customer, the customer may further subnet and assign prefixes to the links in the customer network.

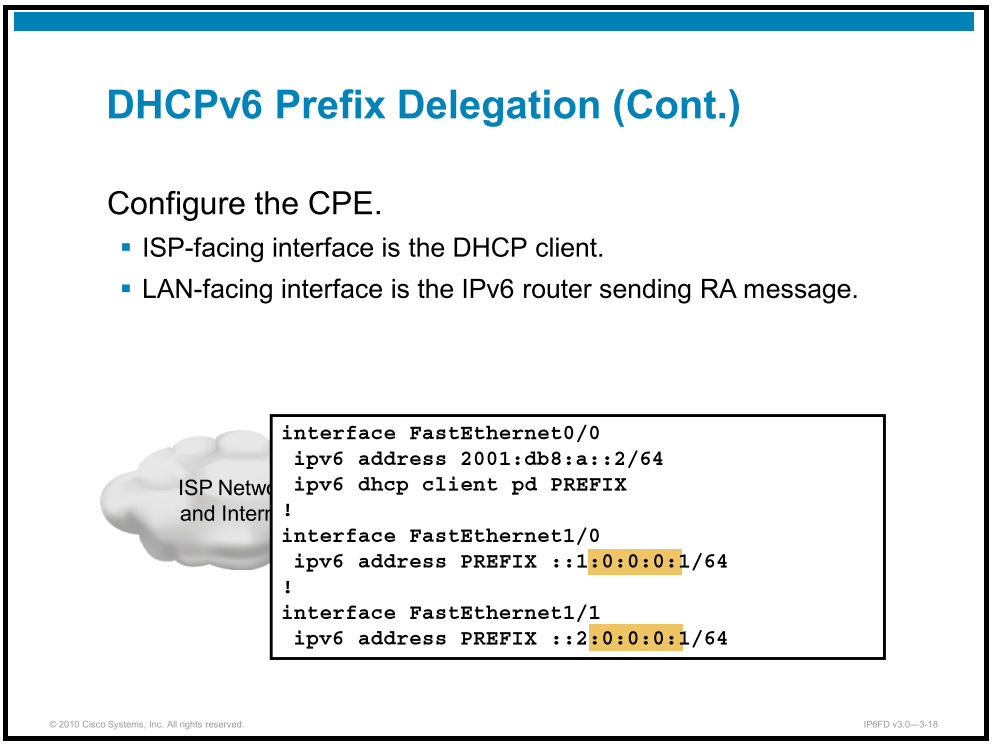


In the figure, the ISP delegating router will act as a DHCP server and will allocate a prefix to the CPE. The CPE will, on one side, act as a DHCP client, acquire the prefix, and then assign smaller prefixes to its own local interfaces. On these interfaces, it will act as an IPv6 router, sending out router advertisements to inform local clients of prefix availability. In this configuration, the ISP indirectly assigns addresses to end nodes.



The figure shows configuration of the DHCP server router.

A DHCP pool, named "Customers" has a prefix-delegation command with a reference to a local pool named C_PREFIX. The local pool C_PREFIX contains specifications mandating that addresses will be allocated to clients, together with prefix length. Addresses that are specified with 2001:db8:c::/40 are reserved for further allocation in blocks of /48. Therefore, each client will receive one /48 from the /40.

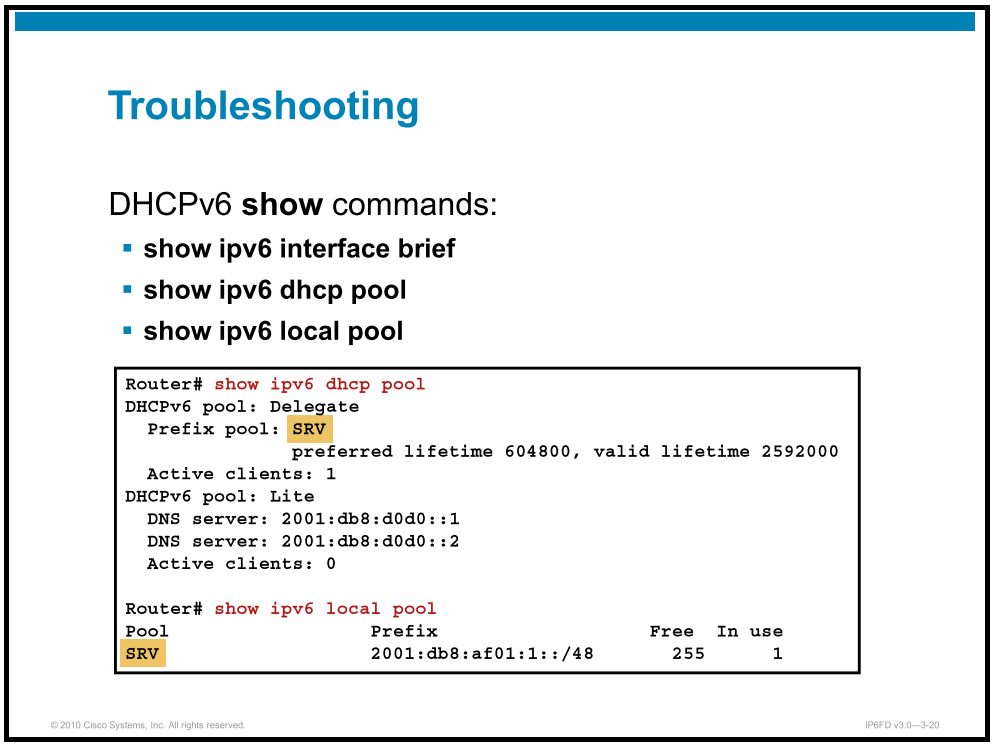


The figure shows a sample configuration of a CPE router. The interface facing the service provider acts as a client and has a prefix delegation reference called *PREFIX*. This prefix delegation will make it possible to refer to the allocated prefix with the variable *PREFIX* later on.

Interfaces facing the LAN devices are configured with an IPv6 address with a reference to the prefix name PREFIX. The ISP defines the first 48 bits, so only the last 80 bits must be listed. For autoconfiguration to work, the network mask is set to /64.

DHCPv6 Troubleshooting

This topic describes how to troubleshoot DHCPv6.



Verification or troubleshooting implies the use of show and debug commands. Use these show commands when checking DHCPv6 operation:

show ipv6 interface brief

This command will list all interfaces and their IPv6 addresses. If one or more interfaces are configured to acquire an IPv6 address through DHCP, you can quickly check if it worked with this command. It will be the first command that you use when troubleshooting client functionality on a router.

show ipv6 dhcp pool

This command will show all DHCPv6 pools on a router. In the figure, the output shows two pools:

- A pool that is named “Delegate” is used for prefix delegation with one client.
- A pool that is named “Lite,” which is used only for sending DNS server information (DHCP Lite).

show ipv6 local pool

Use this command if, when using prefix delegation, you specify the address range with a local pool. The DHCPv6 pool will show only a reference to a local pool, and you will need an additional command to verify the addresses. The figure provides an example.

Troubleshooting (Cont.)

show ipv6 dhcp binding

```
Router# show ipv6 dhcp binding
sinister#sh ipv6 dhcp binding
Client: FE80::C801:4DFF:FE36:8
  DUID: 00030001CA014D360008
  Username : unassigned
  Interface : FastEthernet0/0
  IA PD: IA ID 0x00040001, T1 302400, T2 483840
    Prefix: 2001:db8:af01:1::/56
            preferred lifetime 604800, valid lifetime 2592000
            expires at May 30 2010 01:03 PM (2331577 seconds)
  IA NA: IA ID 0x00040001, T1 0, T2 0
```

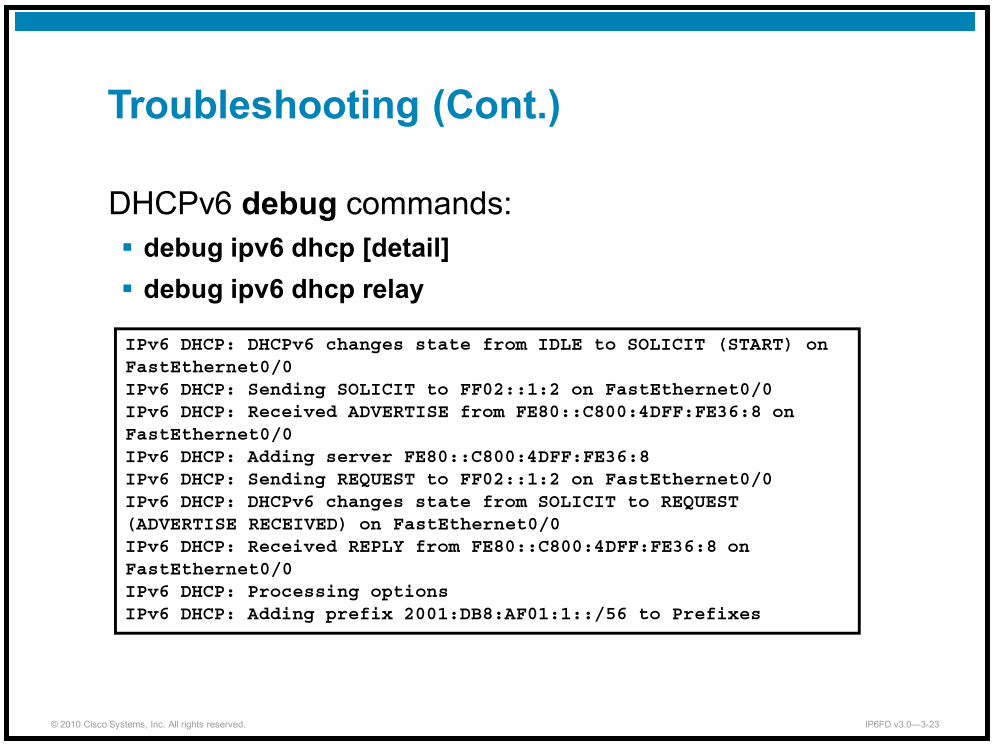

The **show ipv6 dhcp binding** command shows the state of all current clients of the DHCP server. In this figure, you can see the client ID, the interface on which the client is connected, and the assigned prefix.

Troubleshooting (Cont.)

show ipv6 dhcp interface

```
Router# show ipv6 dhcp interface
FastEthernet0/0 is in client mode
  Prefix State is OPEN
  Renew will be sent in 11:39:43
  Address State is IDLE
  List of known servers:
    Reachable via address: FE80::C800:4DFF:FE36:8
    DUID: 00030001CA004D360008
    Preference: 0
    Configuration parameters:
      IA PD: IA ID 0x00040001, T1 302400, T2 483840
        Prefix: 2001:db8:af01:1::/56
                preferred lifetime 604800, valid lifetime 2592000
                expires at May 30 2010 01:03 PM (2331583 seconds)
      Information refresh time: 0
  Prefix name: DelegatedPrefixes
  Prefix Rapid-Commit: disabled
  Address Rapid-Commit: disabled
FastEthernet1/0 is in server mode
  Using pool: Clients
  Preference value: 0
  Hint from client: ignored
  Rapid-Commit: disabled
```


The **show ipv6 dhcp interface** command will tell you whether an interface is in client or in server mode. For client mode interfaces, you will see a list of known servers, the client ID, and acquired information.

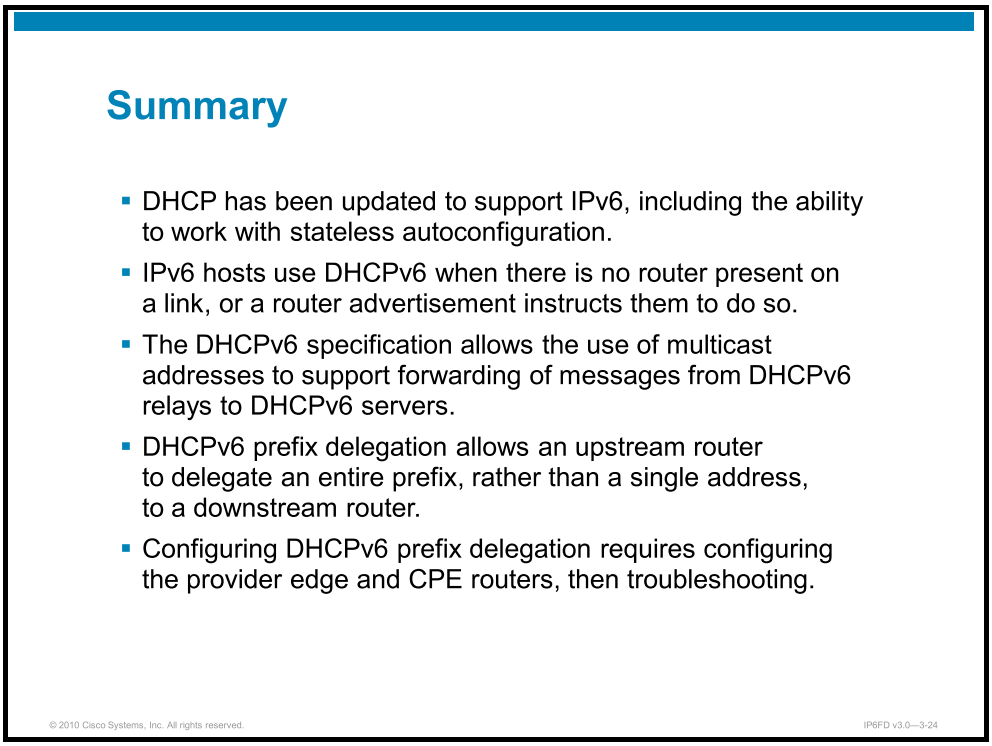


The two debug commands for DHCPv6 are **debug ipv6 dhcp** with the optional **detail** keyword and **debug ipv6 dhcp relay**. The former will be useful when debugging either DHCP server or DHCP client functionality on a Cisco IOS Software router, while the latter will be used when troubleshooting DHCP relay functionality.

The figure lists sample output of the **debug ipv6 dhcp** command on a working DHCP client.

Summary

This topic summarizes the primary points that are discussed in this lesson.



Lesson 4

Understanding QoS Support in an IPv6 Environment

Overview

Not part of the original IP framework, quality of service support is important to many modern Internet technologies, such as unified communications and multimedia services over IP. Understanding the support mechanisms in IPv6 for enabling Quality of Service will help you effectively develop QoS schemes for IP version 6 (IPv6)-enabled networks. This lesson describes quality of service (QoS) support in IPv6 and provides a more detailed explanation of the IPv6 Flow Label.

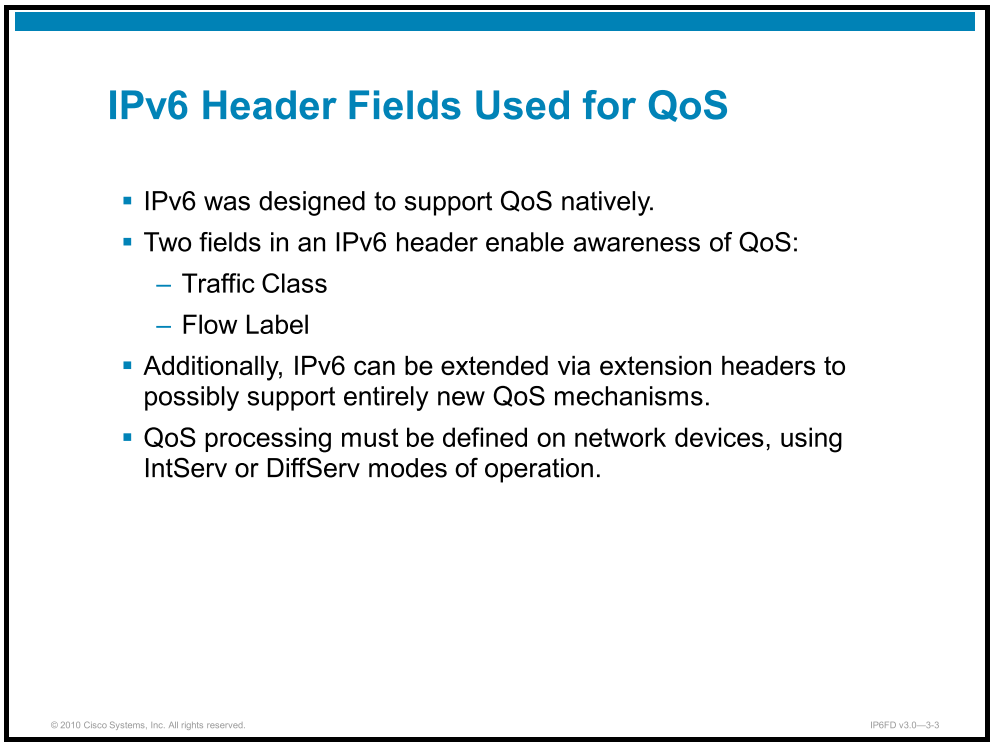
Objectives

Upon completing this lesson, you will be able to describe the fields in the IPv6 header that are used to support QoS and explain how these fields differ from the IP version 4 (IPv4) QoS model. This ability includes being able to meet these objectives:

- Discuss the fields that are used in the IPv6 header to support QoS functions
- Discuss the flow label field in the IPv6 header and how it is structured to potentially support QoS
- Explain how QoS in IPv6 is configured in the Cisco IOS Software

IPv6 Header Fields Used for QoS

This topic describes the IPv6 header fields that are used to support QoS functions.



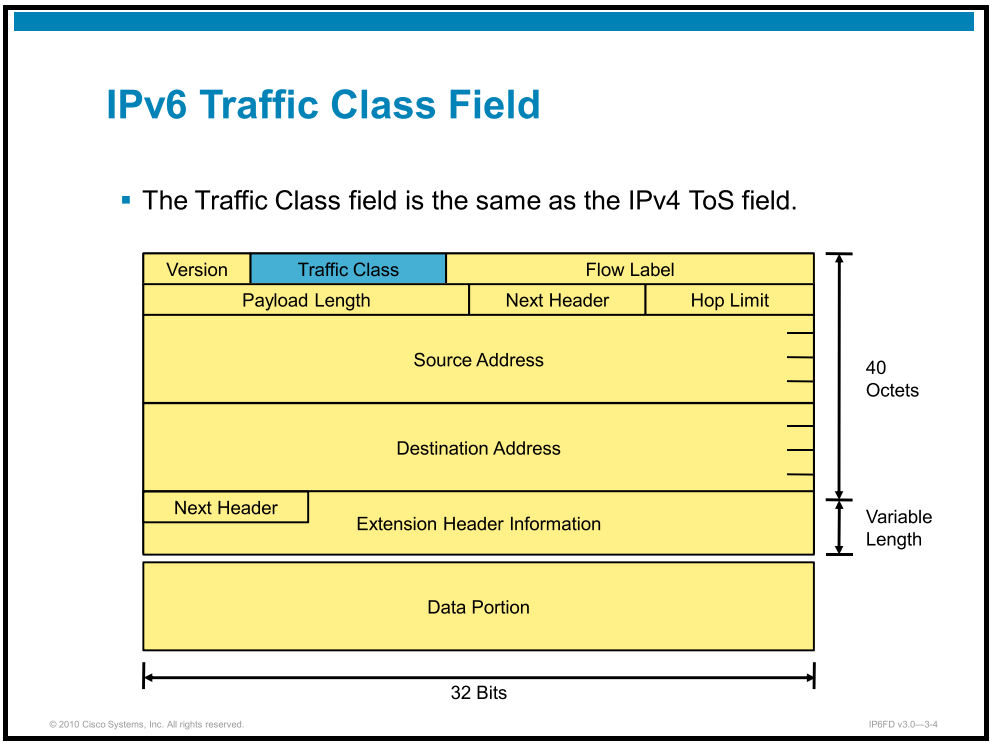
IPv6 was designed to natively support QoS from the beginning. The IPv6 header contains two different fields that are designed to support QoS:

- Traffic Class (8 bits)
- Flow Label (20 bits)

In addition, because of the expanded reach of IPv6 via extension headers, you can add new features to IPv6 by defining new options to put into either the Hop-by-Hop Options header or the Destination Options header. You can also create entirely new extension headers.

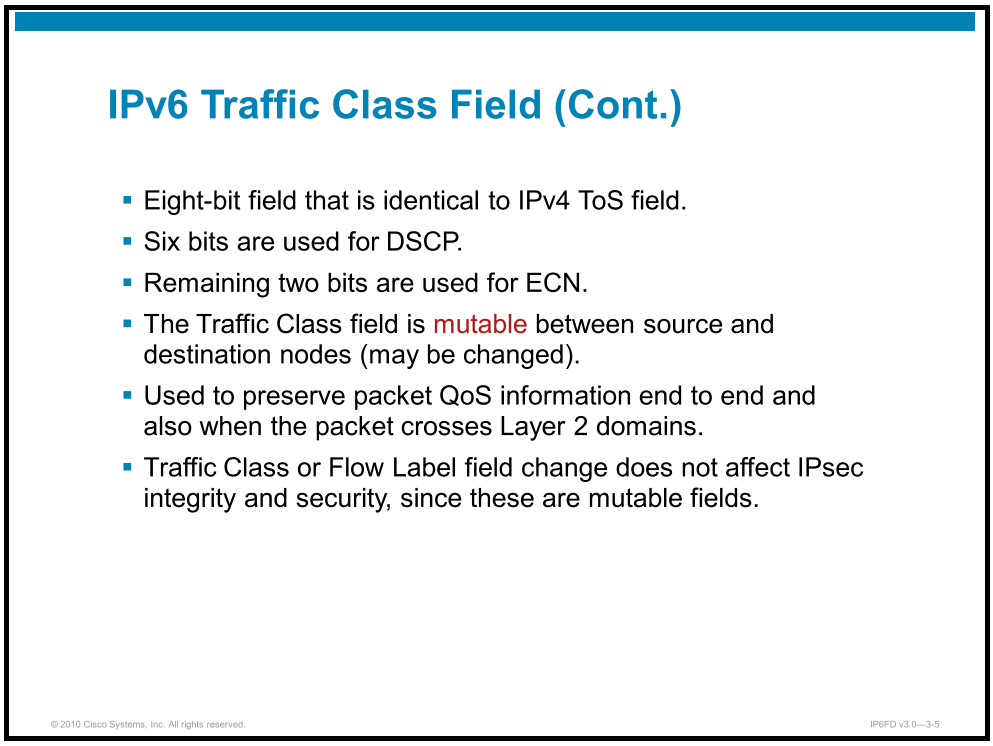
However, entirely new QoS paradigms and mechanisms are not really needed, so current QoS mechanisms are used. Current QoS implementations are based either on the Integrated Services QoS model (IntServ) or on the Differentiated Services QoS model. IntServ is used when absolute QoS guarantees are needed. DiffServ defines “soft” QoS guarantees by just prescribing the behavior of a device that is based on the priority of the packet (the per-hop behavior, or PHB). For example, absolute priority forwarding for important packets and normal forwarding operation for all other packets.

Note You can find more information about QoS mechanisms and principles in the Implementing Cisco Quality of Service (QoS) course. Though the course is focused on IPv4 QoS, the operation of router for IPv6 packets is the same.



The IPv6 Traffic Class field is an eight-bit field identical to the type of service (ToS) field in IPv4.

QoS field position has been moved towards the beginning to allow for easier hardware processing of packets. When receiving a packet, the network device can determine the priority of the packet very early in the process.



Both the IPv6 Traffic Class and IPv4 ToS fields are used in the differentiated services (DiffServ) architecture that is defined in RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. Six bits of the Traffic Class field are specified for use as the differentiated services code point (DSCP) field, in which each DSCP specifies a particular per-hop behavior (PHB) that is applied to a packet at a network device.

The remaining two bits of the Traffic Class field are defined in RFC 3168, *The Addition of Explicit Congestion Notification (ECN) to IP*. ECN is a nonlossy way to indicate congestion on a link and to inform other systems to throttle traffic being sent, to avoid packet drops due to congestion.

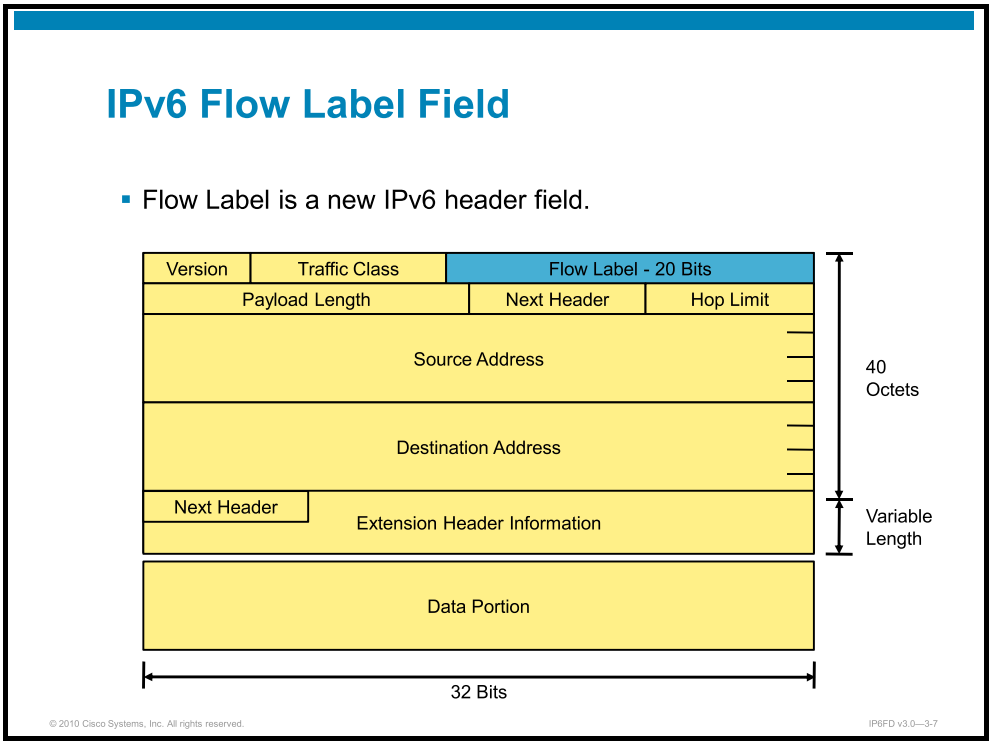
The Traffic Class field resides in the IPv6 packet header and *marks* the packets according to their priority. This means that the Traffic Class can be called a *Layer 3 marker*. Because the IPv6 header does not change in transit, this header is considered an end-to-end marker as well. Layer 2 markers, such as Class of Service (CoS) for Ethernet networks, are valid only for a single Layer 2 domain.

If an IPv6 packet needs to cross multiple Layer 2 domains from its origin to its destination, the Traffic Class marker would be kept unchanged from end to end. New Layer 2 QoS markers would be imposed each time such a packet would be forwarded from one router to another over a Layer 2 network.

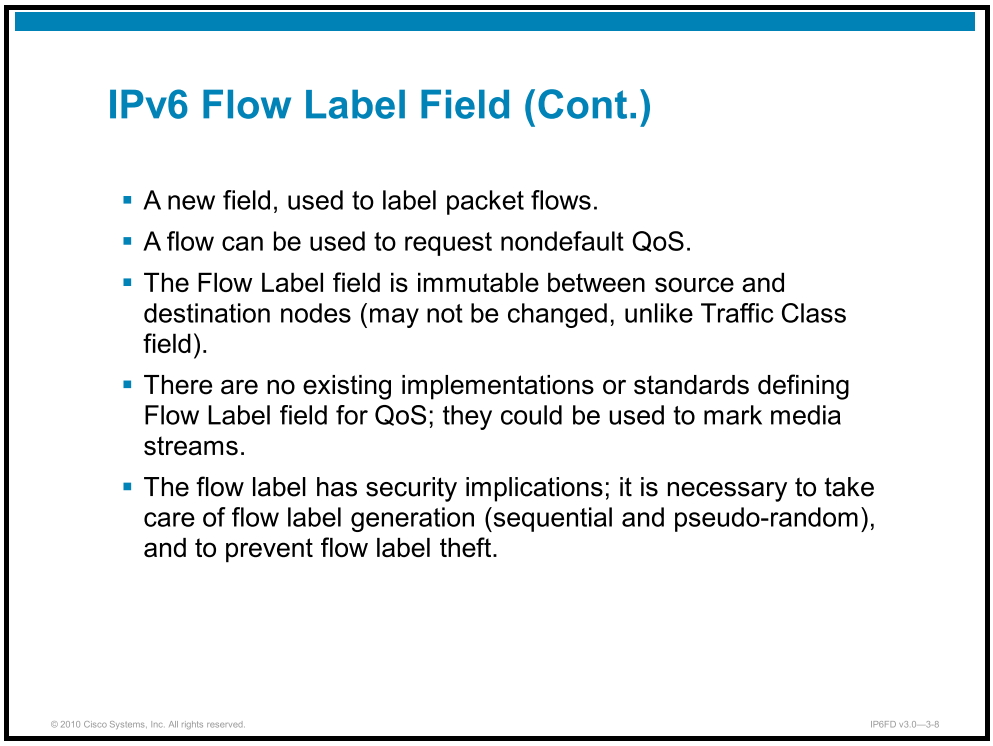
Note	Traffic Class is defined as a mutable field, meaning that it is permissible for intermediate nodes on the path (routers) to change the value of a field during transit. As a result, the Traffic Class field is unprotected by the Authentication Header (AH) when IPsec is in use. This situation is acceptable because the destination node does not discard packets as invalid when, in fact, only a mutable field has changed.

IPv6 and the Flow Label Field

This topic describes the Flow Label field in the IPv6 header and how it is structured to potentially support QoS.



The Flow Label field is a new 20-bit field that appears in the IPv6 basic header.



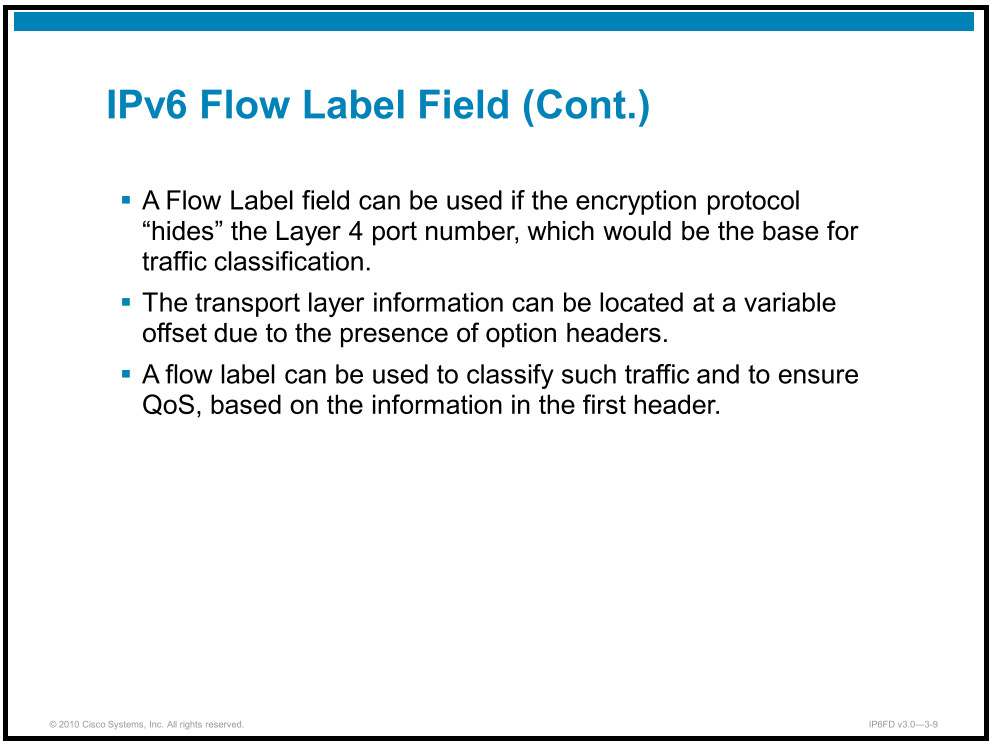
The Flow Label field is used to label packets belonging to specific flows:

- Source address, destination address, and flow label may uniquely identify a flow.
- The flow label can be used for special sender requests, such as nondefault QoS and real-time services.
- There can be multiple flows between a source and destination, as distinguished by separate nonzero flow labels.
- No implementation of the flow label currently exists, nor is its exact usage yet defined. There is a current IETF RFC that describes, at a high level, the basic requirements for the flow label (RFC 3697, *IPv6 Flow Label Specification*).
- The Flow Label field is immutable; its value must arrive at the destination unchanged.

The Flow Label field enables per-flow processing by routers in the path. This function provides differentiation of the traffic at the IP layer without having to open the transport layer header to identify the flow.

Tip Consider a fragmented or encrypted packet. When a packet is fragmented, Layer 4 header information, such as TCP port number, is not carried in each fragment. In this case, for IPv4, QoS cannot be applied to each fragment when QoS classification is based on TCP port numbers. For IPv6 and the flow label, flows are classified only with information in the base header, which appears in each fragment.

For IPsec-encrypted packets, Layer 4 information is encrypted and not available for QoS processing. The flow label and source or destination IP addresses are always visible in an encrypted packet, allowing QoS processing.

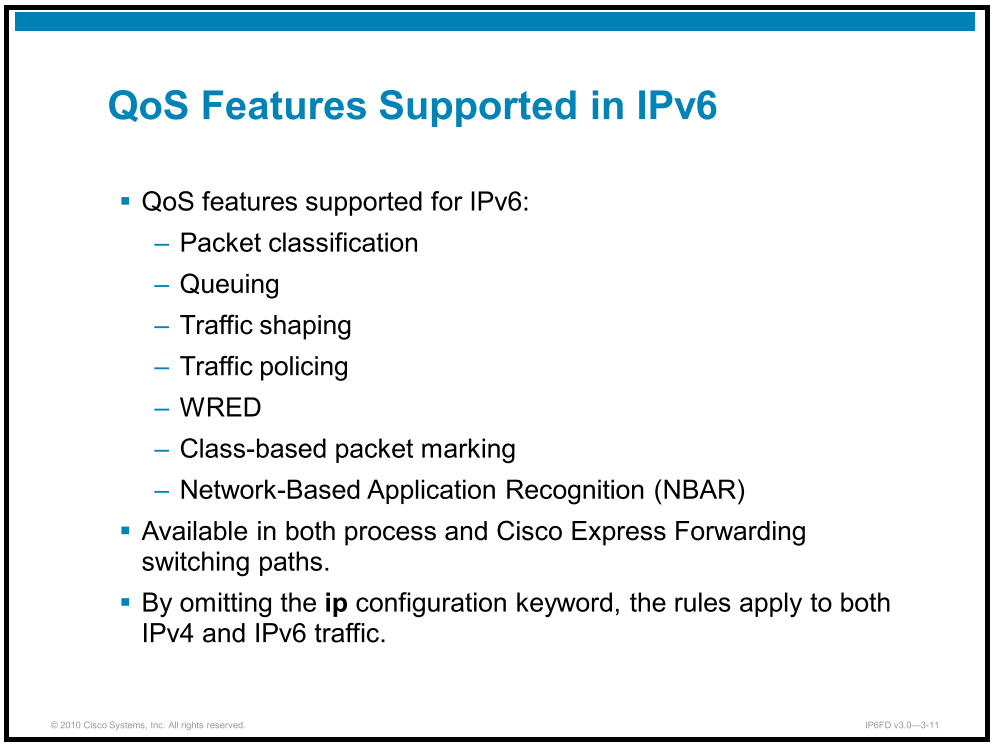


The flow labels can be used to classify traffic if, for some reason, classification cannot be performed otherwise. As an example, packet encryption might obscure the transport layer headers that would otherwise be used for classification. Classification using flow labels could be useful.

Secondly, to enable truly hardware-based QoS processing, the presence—or nonpresence—of option headers changes the position (offset) of a transport layer header. Using flow labels, such packets could be classified based on the information in the IPv6 (Layer 3) header.

IPv6 QoS Configuration

This topic describes Cisco IOS Software and IPv6 QoS configuration and feature support.



Cisco IOS Software supports many IPv4 QoS features for IPv6, with both process and Cisco Express Forwarding switching paths. Classification can be accomplished based on protocol (IPv4 or IPv6) or on protocol-independent values such as DSCP, class of service, or Layer 4 port.

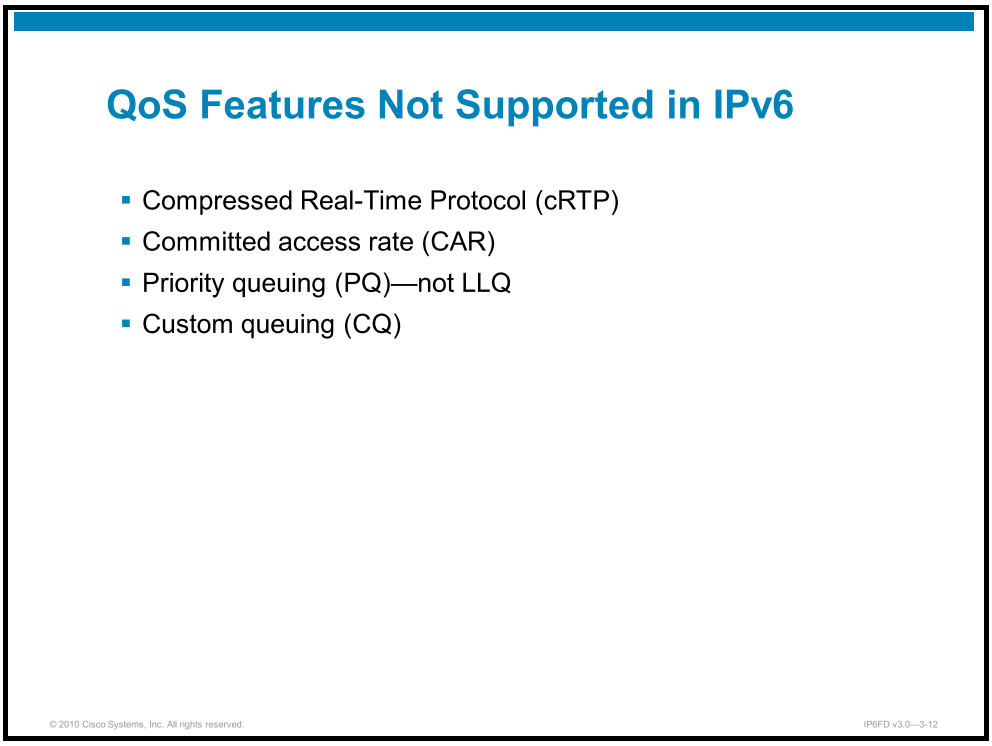
Traffic policing and multiple shaping methods are supported (both Frame-Relay traffic shaping [FRTS] and Generic Traffic Shaping), in addition to weighted random early detection (WRED) congestion avoidance. Most queuing methods are supported, including low latency queuing (LLQ).

With new software platforms, support for Network-Based Application Recognition (NBAR) is becoming available. An example of such a platform is the Cisco ASR 1000 Series Routers; the operating system is Cisco IOS XE Software.

QoS for IPv6 is supported in all newer releases since the Cisco IOS Software Release 12.2T and Cisco IOS Software Release 12.0S.

Using IPv6 in Cisco Express Forwarding hardware switching paths has a few hardware-based restrictions. These restrictions are usually due to ASIC processor cycles for IPv6 addresses (Cisco Catalyst 6500 Series Switches or Cisco 7600 Series Routers). They can also be due to minimum subnet mask size because of hardware TCAM structure (mainly Cisco Catalyst 3560 switches, and so on).

Tip When configuring QoS, the **ip** keyword in the Cisco IOS CLI relates to IPv4. By omitting this keyword, actions are applied to both IPv4 and IPv6 traffic. When using the **ipv6** keyword, QoS actions are applied to IPv6 traffic only.



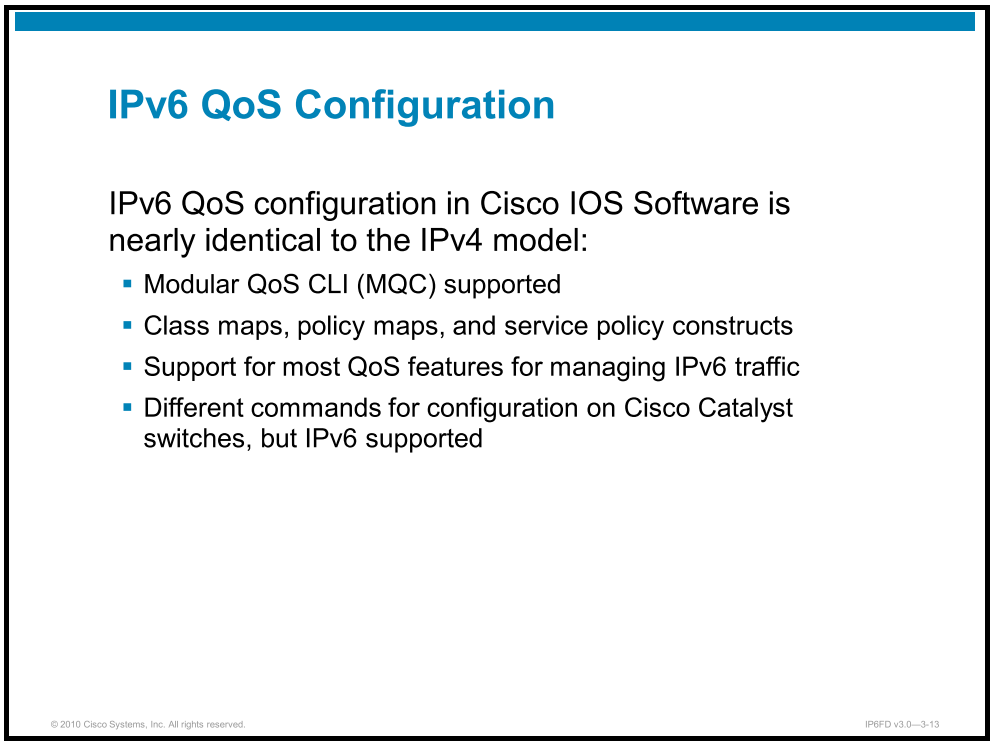
These QoS features are currently not supported in IPv6:

- Compressed Real-Time Protocol (cRTP)
- Committed access rate (CAR)
- Priority queuing (PQ), not LLQ
- Custom queuing (CQ)

Many of these features use older or discontinued software or hardware QoS engines (for example, PQ, CQ, and so on). These engines have been replaced with new QoS engines, and these do support IPv6, in addition to IPv4 (that is, LLQ, class-based weighted fair queuing [CBWFQ], and others).

Features such as NBAR are becoming available for IPv6 as well, especially on software-based routing platforms, such as the Cisco ASR 1000 Series Routers.

Technologies such as cRTP have been used in IPv4 networks in the past when bandwidth was less available. There was no demand to implement such features for IPv6.



Configuring IPv6 QoS in Cisco IOS Software is nearly identical to configuring QoS for IPv4. These basic configuration elements are the same:

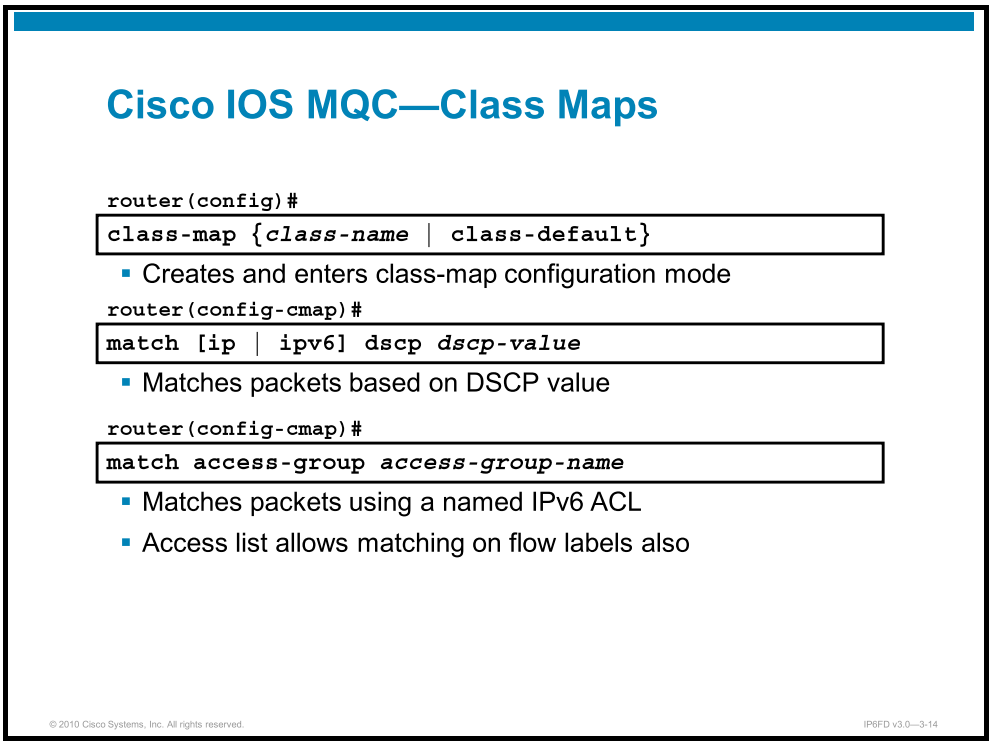
- Modular QoS CLI (MQC) is supported.
- Class maps, policy maps, and service policy commands are used.

The MQC is a consistent, flexible way to configure QoS policies on Cisco routers. The three elements in the MQC ease configuration tasks by enabling entire QoS policies to be applied to interfaces using a single command, as follows:

- **Class maps:** Class maps define which traffic to apply QoS policies to. Powerful classification commands are available to sort traffic that is based on Layer 2, 3, 4, or 7 criteria. New for IPv6 is the ability to match traffic that is based on protocol, so, for example, you can match just IPv6 DSCP Expedited Forwarding (EF) traffic.
- **Policy maps:** Policy maps are used when QoS policies are applied to the traffic placed in the class maps previously defined. Packet marking (Layer 2 or 3), policing and shaping, congestion avoidance, WRED, and various queuing methods are applied using policy maps. In addition, CBWFQ bandwidth guarantees can be applied, and priority queues can be established for real-time traffic.
- **Service policies:** Service policies apply policy maps to interfaces in a specific direction (inbound or outbound).

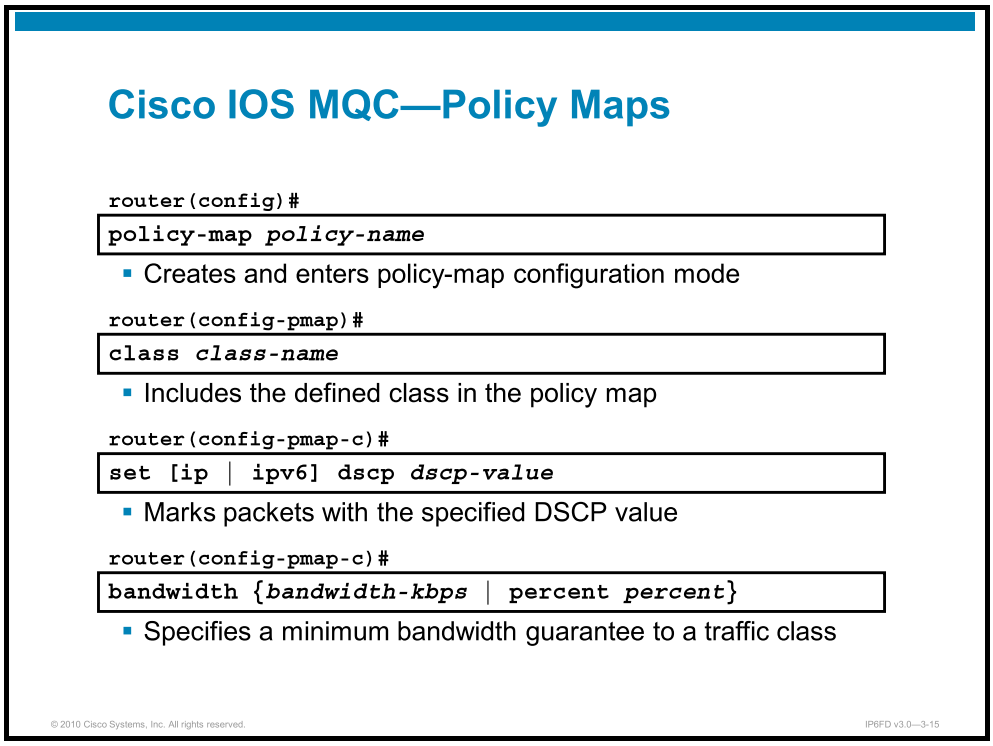
Configuration of QoS on Cisco Catalyst switches uses different commands or subcommands, such as **set-dscp-transmit** or **policed-dscp-transmit** when using traffic policing. The policing mechanism on Catalyst switches is configured in the same way as for IPv4 because it works for both IPv4 and IPv6 traffic.

Note On Cisco Catalyst switches, more stress is on Layer 2 QoS, which depends on the CoS marker to determine hardware queues for the traffic.



Defining class maps is the first step in creating a QoS policy using the MQC. For each class of traffic that you wish to apply policies to (including a default class), decide which traffic to place in that class. Classification can be done using any of these methods:

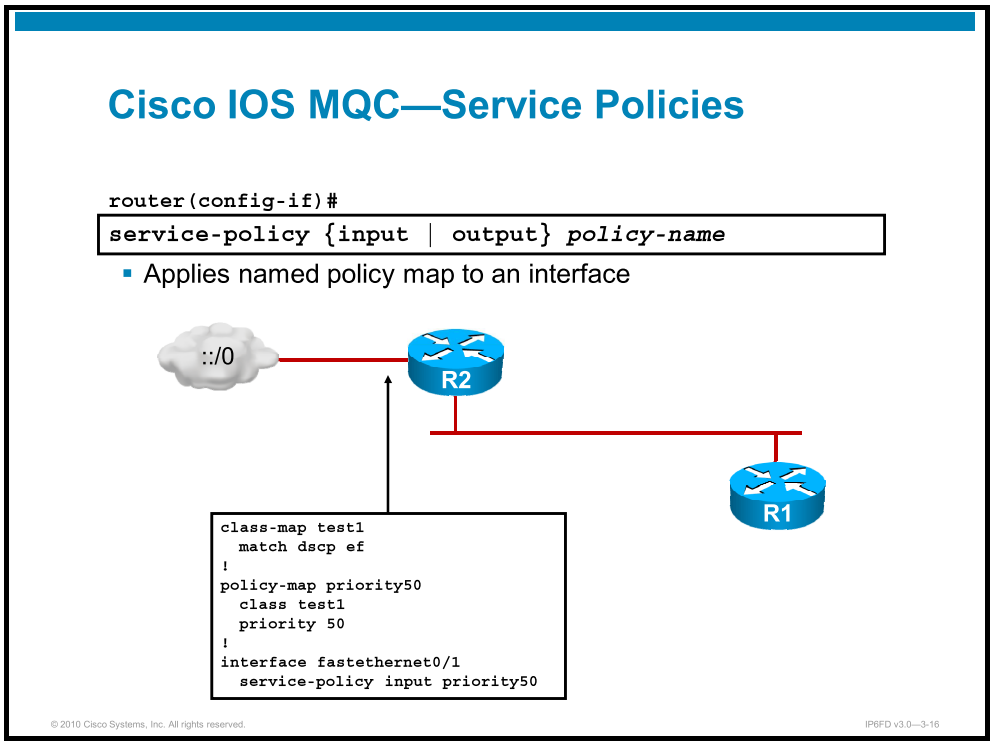
- Layer 3 IPv6 source or destination address
- Any other Layer 3 or Layer 4 criteria, when using an access list
- Layer 2 class of service (CoS)
- IPv6 DSCP, stored in the Traffic Class field
- Protocol type
- Flow label. Flow labels can be matched using an access list (using the **permit ipv6 any any flow-label** statement).



The policy map is where the actual QoS policy is applied to traffic on Cisco routers. Multiple policy maps may be created, each referencing a different set of class maps. In the policy map, the **class** command is used to identify each class map that you wish to apply QoS mechanisms to. The actions under the **pmap-c** configuration mode fall into these general categories:

- **Marking:** These attributes can be set and modified:
  - Cell loss priority (CLP) bit when a packet is due to be transferred over an ATM network
  - CoS value of an outgoing packet on Ethernet networks
  - DSCP value in the Traffic Class field
  - Discard eligible (DE) bit setting in the address field of a Frame Relay frame
  - Multiprotocol Label Switching (MPLS) experimental (EXP) field on all imposed label entries
  - MPLS EXP field value in the topmost label on either an input or an output interface
- **Policing and shaping:** These traffic regulation mechanisms are supported:
  - Generic traffic shaping
  - Class-based traffic shaping
  - Frame Relay traffic shaping (FRTS)
- **Congestion avoidance:** These types of WRED are supported:
  - Flow-based WRED
  - DSCP-based WRED

- **Congestion management:** This category comprises the various queuing methods and supports the following:
  - Weighted fair queuing (WFQ)
  - CBWFQ
  - LLQ, which is really CBWFQ plus a strict priority queue



The **service-policy** command applies a given policy map to a specific interface, in either the inbound or outbound direction. Only one policy map per direction can be configured at a time on an interface.

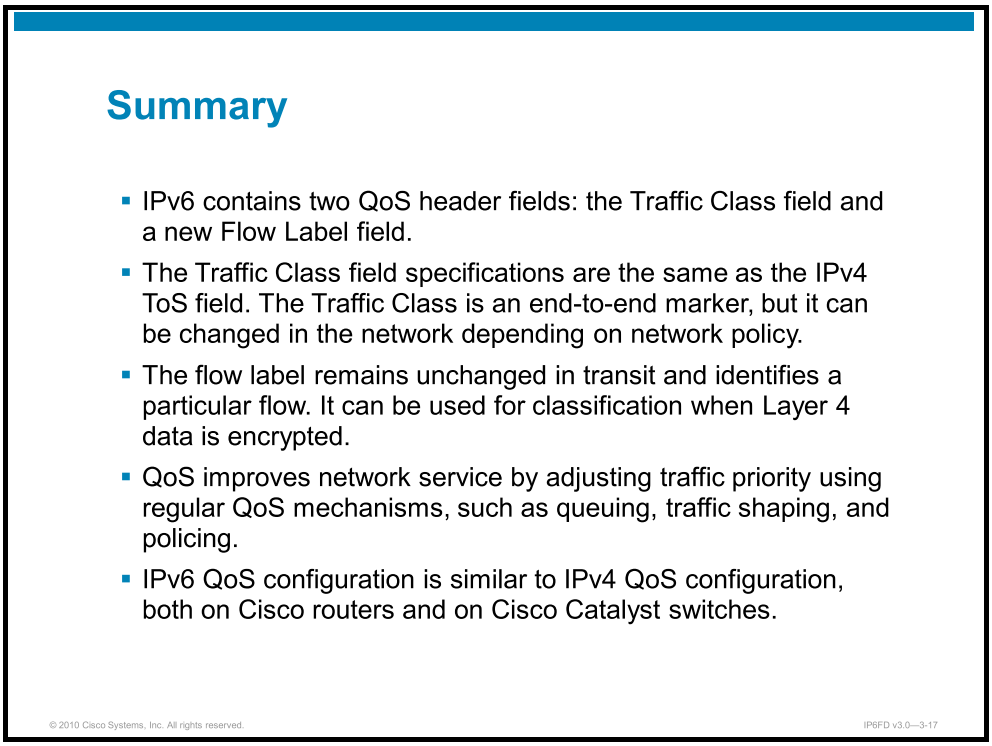
The figure shows a complete, although simple, example of a QoS configuration using the MQC, as follows:

- The **class-map test1** command matches packets with the DSCP value of EF (usually real-time traffic).
- The **policy-map priority 50** command includes a single class, test1, and assigns the packets in the class to a priority queue with a minimum bandwidth guarantee of 50 kb/s.
- The priority50 policy map is applied to FastEthernet 0/1 in the inbound direction with the **service-policy** command.

Note The DSCP value that is described as EF is actually recommended to be “101110.” The “EF” is not intended to be a hexadecimal value, but stands for “Expedited Forwarding.” It is designed to provide low-delay, low-jitter, and low-loss service. A classic use of EF is to support VoIP, which requires these two characteristics to perform well: low latency and low jitter. VoIP is not very sensitive to loss of an occasional packet.

Summary

This topic summarizes the primary points that are discussed in this lesson.



Resources

To learn more about IPv6 QoS configuration, please refer to the following material:

- *Cisco IOS IPv6 Configuration Guide, Release 12.4: Implementing QoS for IPv6* at http://conft.com/en/US/docs/ios/ipv6/configuration/guide/ip6-qos.html#wp1055789

Lesson 5

Using Cisco IOS Software Features

Overview

Cisco supports many tools and applications that manage and troubleshoot networks. To maintain this capability for IP version 6 (IPv6)-enabled networks, these tools and applications have been modified. This lesson describes Cisco IOS tools such as Telnet, TFTP, Secure Shell Protocol (SSH), and others.

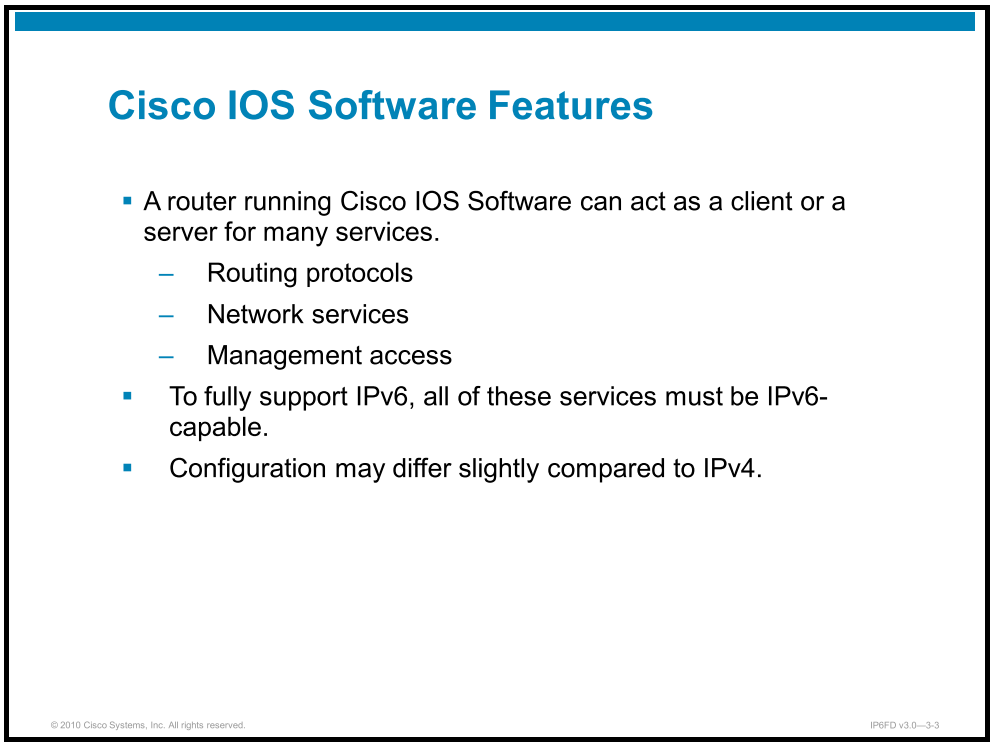
Objectives

Upon completing this lesson, you will be able to describe and configure advanced Cisco IOS features to support IPv6. This ability includes being able to meet these objectives:

- Describe and configure SSH and Telnet on Cisco routers
- Describe and configure TFTP, HTTP, traceroute, ping, and NTP on Cisco routers
- Describe Cisco Discovery Protocol support for IPv6
- Describe Cisco Express Forwarding support for IPv6
- Describe IP SLA functionality for IPv6

Cisco IOS Software Features

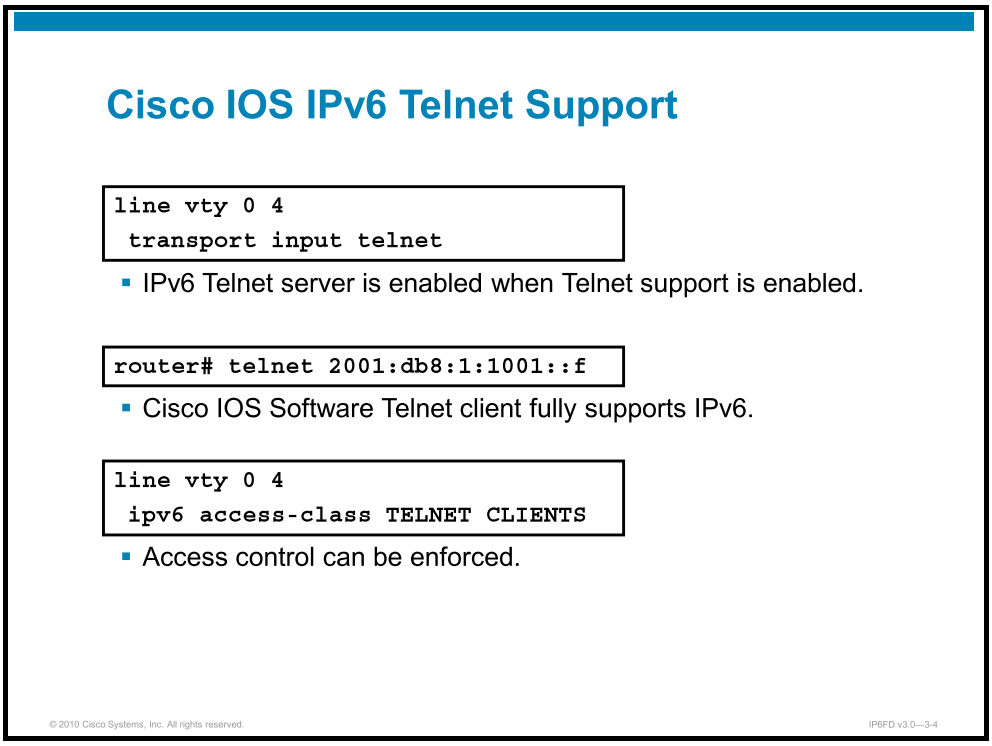
This topic describes IPv6 service support in Cisco IOS Software.



A router running Cisco IOS Software is not just forwarding packets. It also runs routing protocols, which may need to communicate over IPv6. (The “IPv6-Enabled Routing Protocols” module specifically addresses IPv6 routing protocols.) In addition, a router can also offer various network services, can act as a troubleshooting platform, and must support various protocols for administrative access.

In an IPv6-only network, all these features must fully support IPv6 as a transport mechanism.

In most cases, IPv6 configuration commands do not differ significantly from their IPv4 counterparts. Sometimes the commands are the same. In some cases, however, configuring IPv6 and configuring IPv4 are substantially different.



Telnet is a primary method of doing remote management. Unfortunately, Telnet is insecure because Telnet data goes over the network in cleartext.

The Telnet client and server support IPv6 connections. You can connect directly to a router using an IPv6 Telnet client. You can also initiate an IPv6 Telnet connection from a router.

These commands enable Telnet access on a Cisco router:

- router(config)#**line vty 0 4**
- router(config-line)#**password** *password*
- router(config-line)#**login local**

Telnet protocol can be explicitly enabled or disabled with a **[no] transport input telnet** line configuration command.

You can restrict access via Telnet by applying an access control list (ACL) on the virtual terminal interface, an ACL on the ingress router interface, or both.

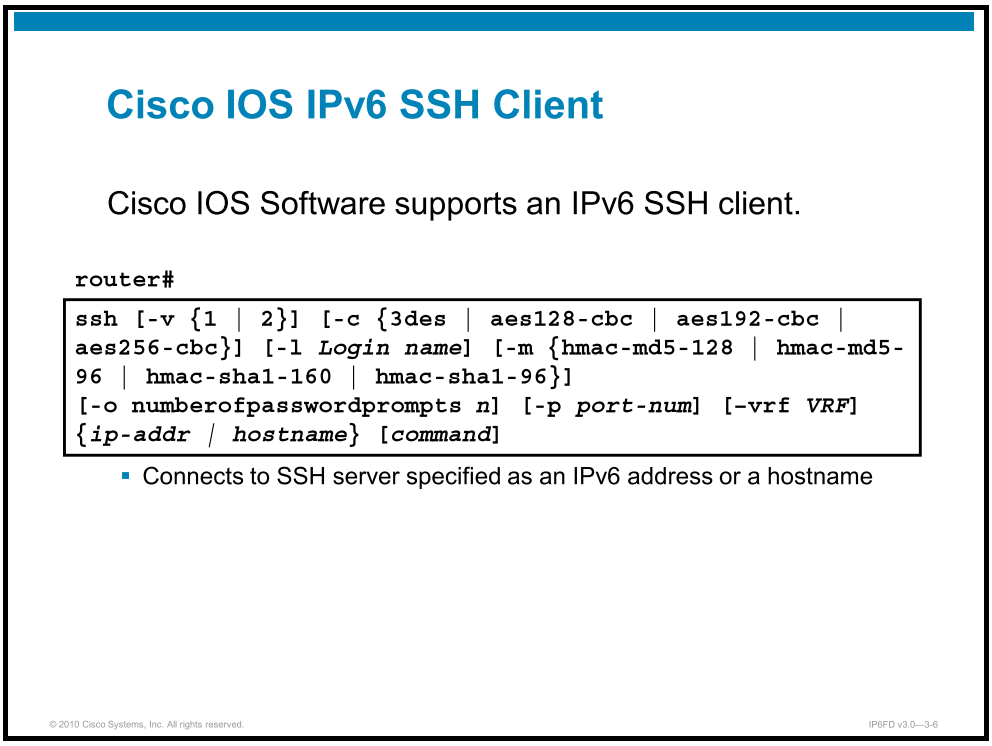
Cisco IOS IPv6 SSH Support

- Cisco IOS Software supports IPv6 SSH:
  - Can replace Telnet for interactive session on router for added security
    - Authentication is protected.
    - Session is protected.
- SSH is available in Cisco IOS Software with 3DES cryptographic software.

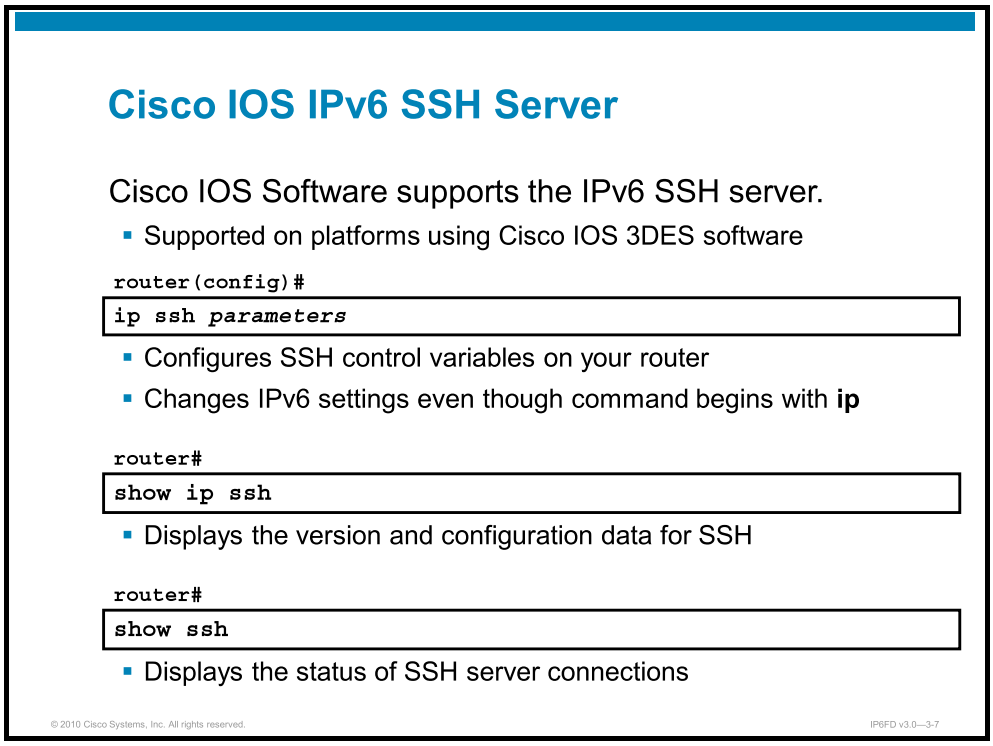
SSH is a popular replacement for Telnet because it provides security features not available on Telnet. In a plain Telnet user session, the authentication and the session are transported in cleartext over the network. Anyone in the path between the user and the router can intercept the session information.

SSH protects the interactive session through encryption and can also be used to provide stronger authentication mechanisms and other features. SSH is available for both IPv4 and IPv6 when running an IPsec-capable version of Cisco IOS Software.

You can restrict access via SSH by applying an ACL on the virtual terminal interface, an ACL on the ingress router interface, or both.



In Cisco IOS Software, the SSH IPv6 client is supported. An IPv6 address or a hostname can be used as a destination in an SSH command. The command syntax is the same as the IPv4 command syntax, and many of the command switches listed in the figure are optional.



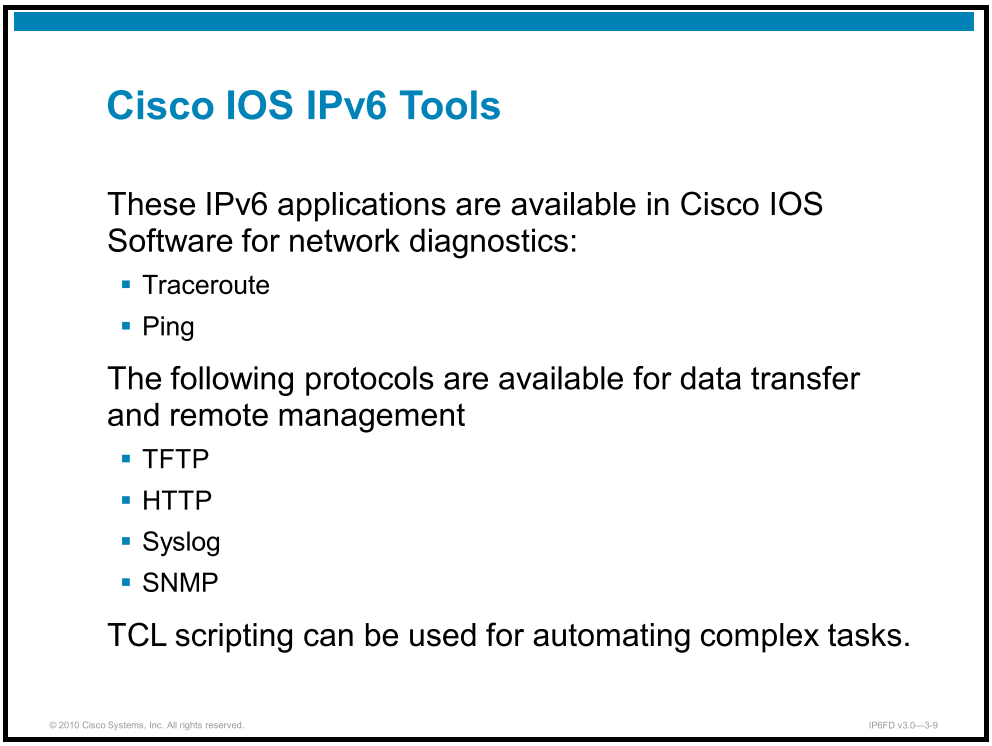
The commands for the SSH server support IPv6 on all supported platforms running IPsec Cisco IOS Software code.

The router must have already generated a key pair. The command to generate a key pair is **router(config)# crypto key generate rsa**. This command, in turn, requires that you have a hostname and a domain name that are defined by using these commands:

- router(config)#**hostname somerouter**
- router(config)#**ip domain name example.com**

Cisco IOS IPv6 Tools

This topic describes three common network diagnostic tools available in Cisco IOS Software.



These IPv6-enabled applications are available in Cisco IOS Software:

- **traceroute** *destination*: This command accepts a destination IPv6 address or hostname as an argument and generates IPv6 traffic to report each IPv6 hop that is used to reach a destination address.
- **ping** *destination*: This command is used to send an Internet Control Message Protocol, version 6, (ICMPv6) echo request to a destination. The ICMPv6 echo reply is reported on the console. Extended ping is also supported.

These applications can take as argument an IPv6 address or a hostname that will resolve to an IPv6 address. (Currently, Domain Name System [DNS] AAAA records are supported, but the A6 record is not.)

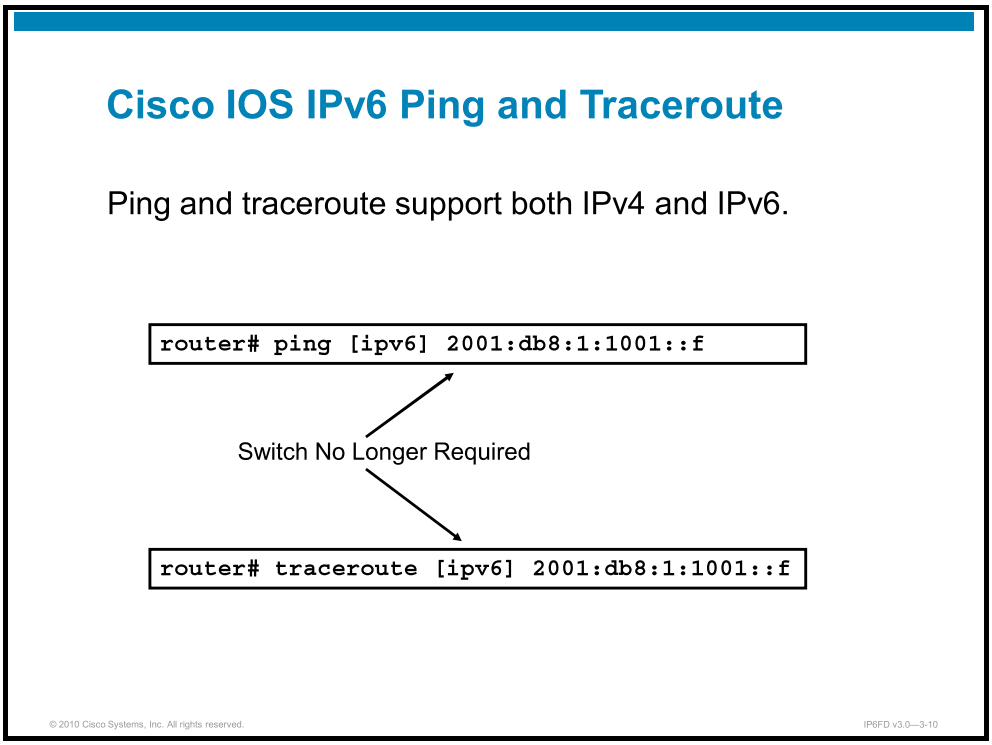
Microsoft Internet Explorer does not currently support the use of literal IPv6 addresses (“literals”) as URLs, although IPv6 hostnames do function. Internet Explorer 7 and the current release of Mozilla Firefox do support literals.

The following protocols are used in Cisco IOS Software for network management:

- **TFTP:** Used for sending or receiving files, most often configuration settings or software binaries. Transparently supports IPv6 addresses. A router can act as a client or as a server.
- **HTTP:** You can use a web browser to connect to a router over IPv6. The configuration page that is shown in the browser window will be the same as that for web-based connections made over IPv4. Traffic filtering via ACLs with the **http access-class** command is *not* currently supported for IPv6-based HTTP and HTTPS connections. An ACL on the ingress router interface would need to restrict this traffic. The HTTP service can be disabled for both IPv4 and IPv6 by issuing the **no ip http server** command.

- **Network Time Protocol (NTP):** Used for time synchronization. Correct time can be very important when comparing logs from different devices and when certificates are used for authentication. NTPv4 adds support for IPv6.
- **Simple Network Management Protocol (SNMP):** Used for remote management and reporting. Often used for status monitoring. SNMP fully supports IPv6 as transport protocol.
- **Syslog:** Used for sending log messages to a remote host over User Datagram Protocol (UDP). Syslog in Cisco IOS Software supports IPv6 as transport protocol.

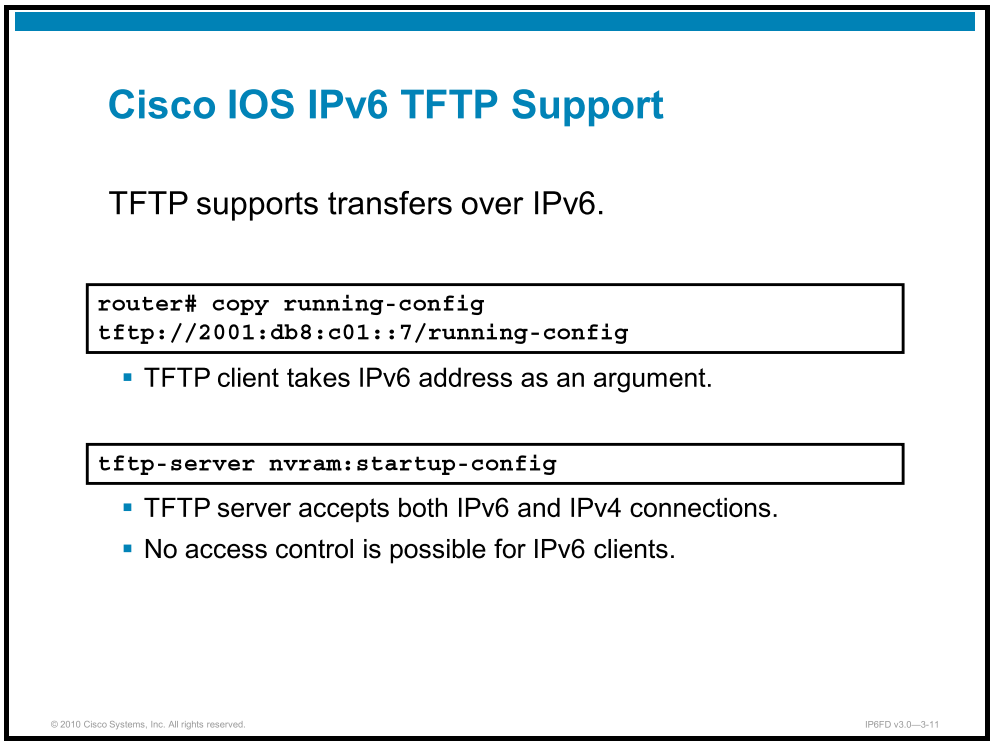
Tool Command Language (Tcl) is a scripting language that is implemented in Cisco IOS Software and can be used for automating tasks. It can integrate with Embedded Syslog Manager (ESM), Cisco IOS Embedded Event Manager (EEM), and interactive voice response (IVR).



The Cisco **ping** command is multiprotocol-enabled and will send either IPv4 ICMP or IPv6 ICMPv6 messages that are based on the address that is being pinged. The IPv6 behavior of the **ping** command is much the same as it is for IPv4.

Tip	Ping is an example of a dual-stack application. Instead of having a **ping** command for IPv4, and a **ping6** command for IPv6, a single ping examines the user input, parses the command line, and sends packets out over either IPv4 or IPv6.

The **traceroute** command is also dual-stacked. In both cases, the **ipv6** parameter is optional and no longer required. The application will detect the IPv6 address and use the appropriate protocol.

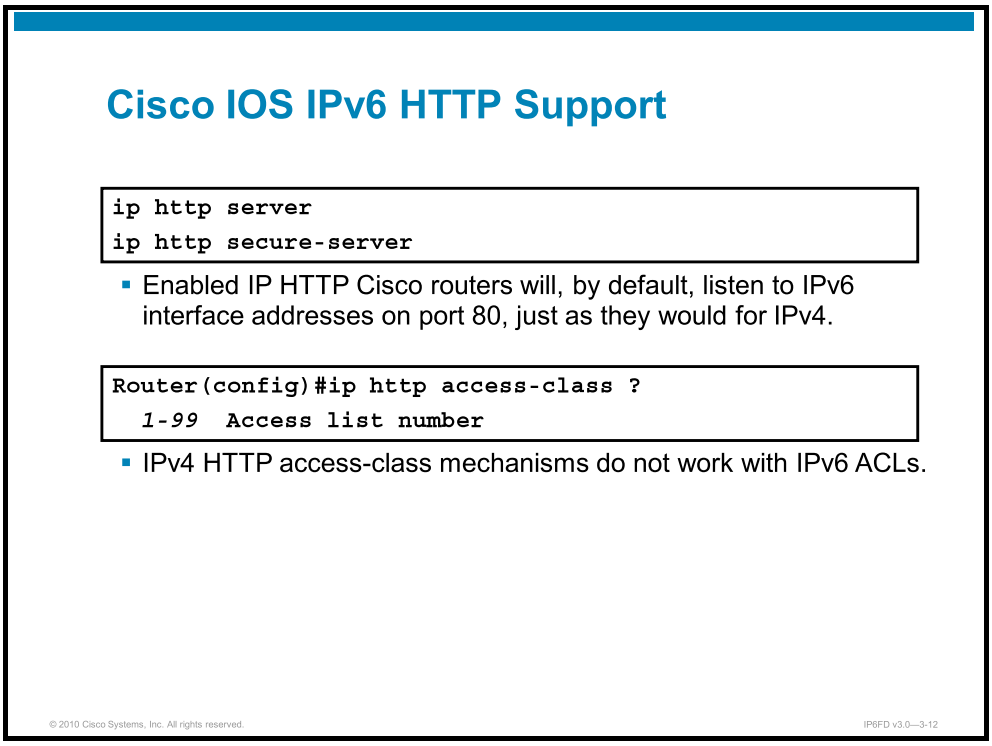


TFTP file download and upload can be done via IPv6. For example, to save the running configuration of the router to an IPv6 TFTP server, use this command:

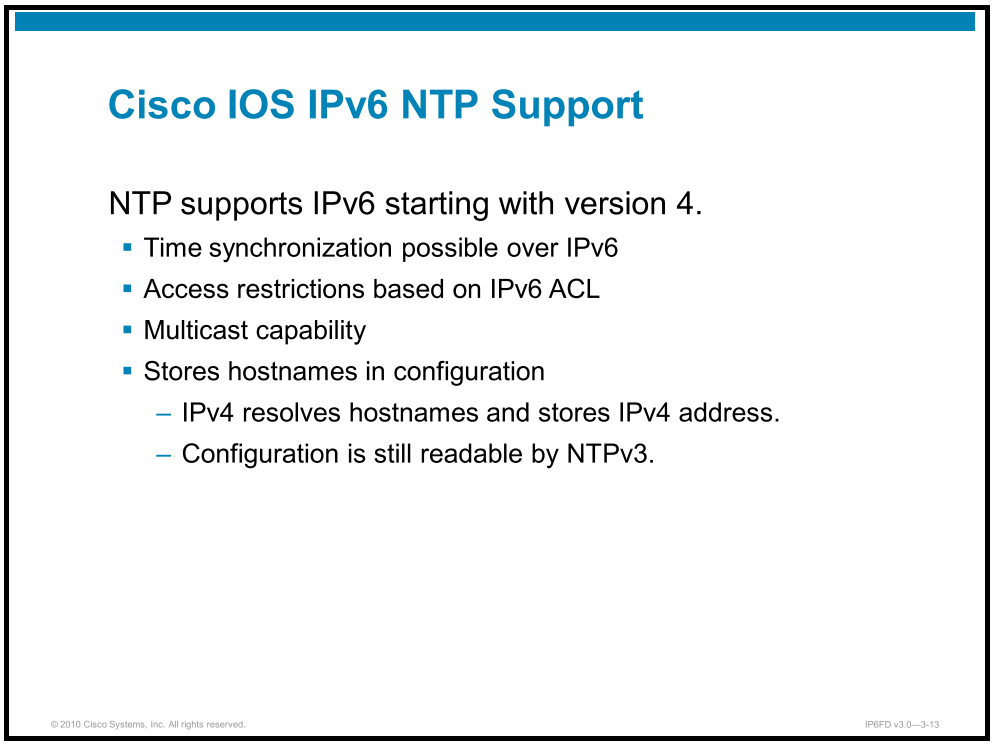
copy running-config tftp://2001:db8:c01::7/running-config

The address of the TFTP server, where the configuration will be saved in a running configuration file, is 2001:db8:c01::7.

The TFTP server is started with **tftp-server** *device*:*filename*. It starts both IPv4 and IPv6 service. While IPv4 access can be restricted with an ACL, IPv6 access cannot. You need to configure an ACL, which restricts TFTP UDP packets on an interface.



Enabling the HTTP server on a Cisco router will cause that router to listen to port 80 on all IPv4 and IPv6 addresses. Security can be applied to an IPv4 server via the **access-class ACL** command. The **access-class ACL** command is not currently supported for IPv6 HTTP or HTTPS service, so this command must be secured by applying the appropriate ACLs on the ingress router interfaces.



NTP is designed to time-synchronize a network of machines. NTP runs over UDP, which in turn runs over IPv4. NTP version 4 (NTPv4) is an extension of NTP version 3. NTPv4 supports both IPv4 and IPv6 and is backward-compatible with NTPv3.

NTPv4 provides the following capabilities:

- NTPv4 supports IPv6, making NTP time synchronization possible over IPv6.
- Security is better than that of NTPv3. The NTPv4 protocol provides a whole security framework that is based on public key cryptography and standard X509 certificates.
- Using specific multicast groups, NTPv4 can automatically calculate its time-distribution hierarchy through an entire network. NTPv4 automatically configures the hierarchy of the servers to achieve the best time accuracy for the lowest bandwidth cost. This feature leverages site-local IPv6 multicast addresses.

NTPv4 works in much the same way as does NTP. An NTP network usually gets its time from an authoritative time source, such as a radio clock or an atomic clock that is attached to a time server. NTP then distributes this time across the network. NTP is extremely efficient. No more than one packet per minute is necessary to synchronize two machines to the accuracy of within a millisecond of each other.

NTP uses the concept of a “stratum” to describe how many NTP “hops” away a machine is from an authoritative time source. A stratum 1 time server typically has an authoritative time source (such as a radio or atomic clock, or a Global Positioning System [GPS] time source) directly attached. A stratum 2 time server receives its time via NTP from a stratum 1 time server, and so on.

NTP avoids synchronizing to a machine whose time may not be accurate in two ways. First, NTP never synchronizes to a machine that is not in turn synchronized itself. Second, NTP compares the time that is reported by several machines and will not synchronize to a machine whose time is significantly different from the others, even if its stratum is lower. This strategy effectively builds a self-organizing tree of NTP servers.

The Cisco implementation of NTP does not support stratum 1 service. In other words, it is not possible to connect to a radio or an atomic clock (for some specific platforms, however, you can connect to a GPS time-source device).

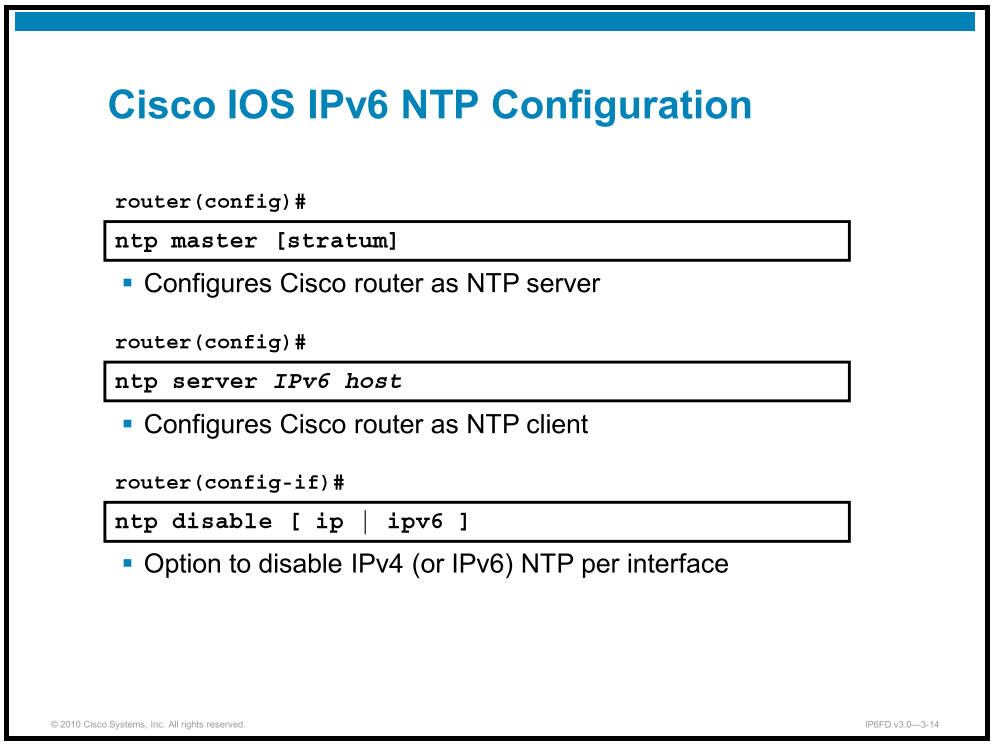
If the network is isolated from the Internet, the Cisco implementation of NTP allows a machine to be configured so that it acts as though it is synchronized via NTP, when it really has determined the time using other means. Other machines can then synchronize to that machine via NTP.

A number of manufacturers include NTP software for their host systems, and a publicly available version for systems running UNIX and its various derivatives is also available. This software also allows UNIX-derivative servers to acquire the time directly from an atomic clock, which would then propagate time information along to Cisco routers.

The communications between machines running NTP (known as "associations") are usually statically configured. Each machine is given the IPv4 or IPv6 address of all machines with which it should form associations. Accurate timekeeping is made possible by exchanging NTP messages between each pair of machines with an association.

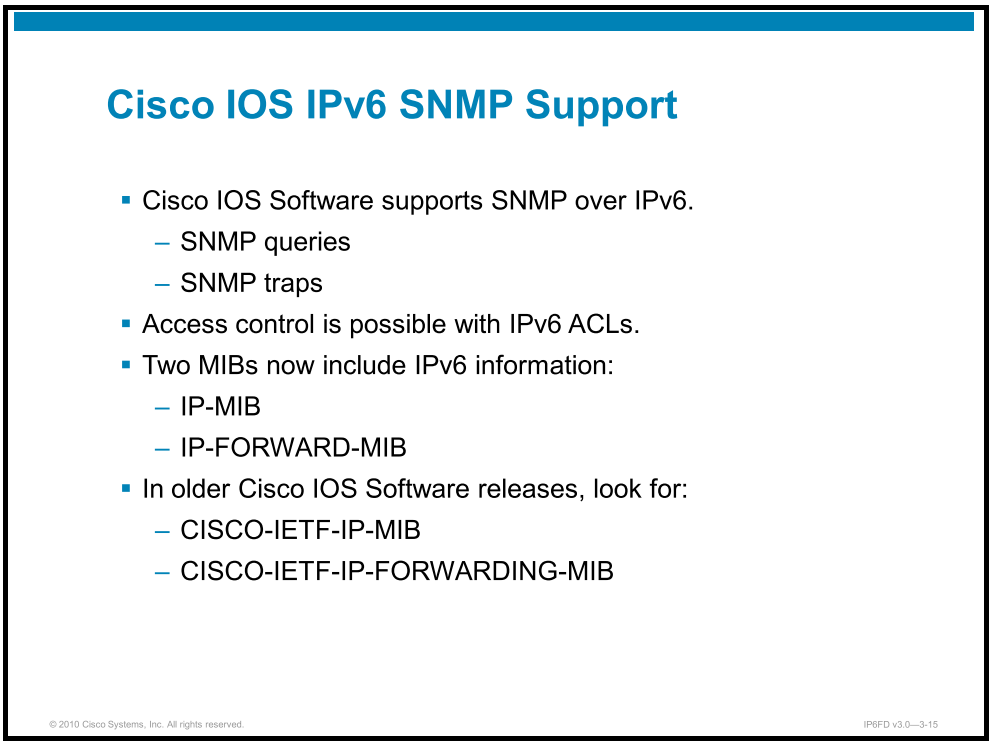
NTPv3 supports sending and receiving clock updates using IPv4 broadcast messages. Many network administrators use this feature to distribute time on LANs with minimum client configuration. For example, Cisco corporate LANs use this feature over IPv4 on local gateways. End-user workstations are configured to listen to NTP broadcast messages and synchronize their clocks accordingly.

In NTPv4 for IPv6, IPv6 multicast messages instead of IPv4 broadcast messages are used to send and receive clock updates.



The **ntp master** command starts the NTP server on the Cisco IOS Software router. It allows NTP to operate over both IPv4 and IPv6. To completely disable IPv4 NTP, the administrator needs to disable NTP on every interface with the command **ntp disable ip**. Similarly, it is possible to completely disable NTP with **ntp disable** or disable IPv6 NTP support with **ntp disable ipv6**.

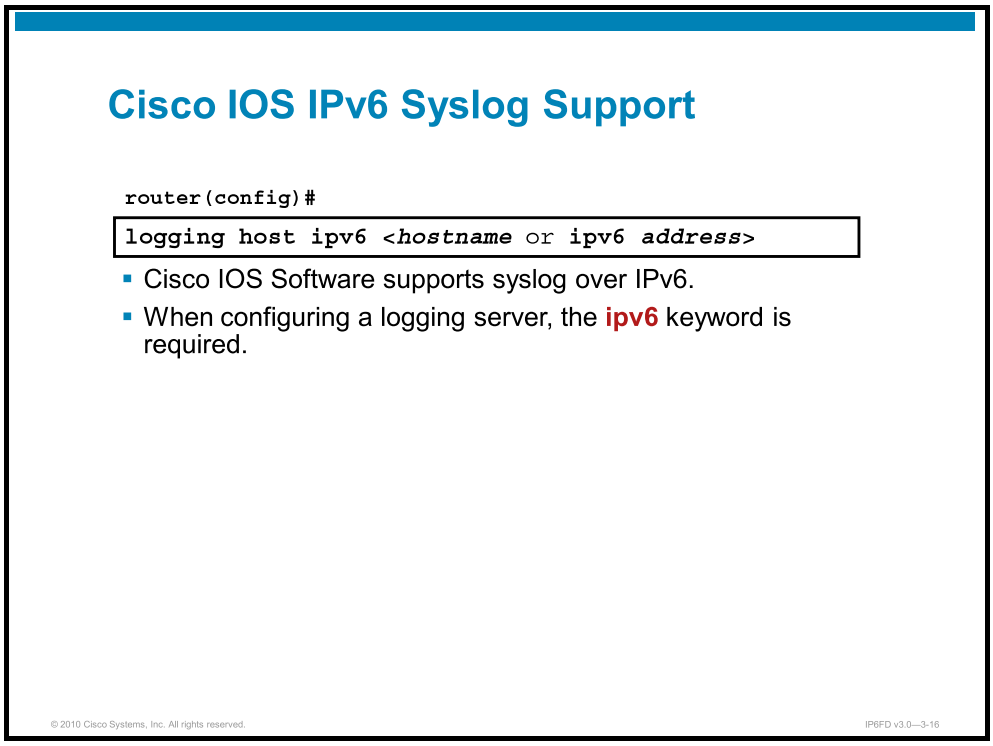
To synchronize Cisco IOS Software to an external clock source, you must use the **ntp server** command. NTPv4 adds DNS support for IPv6. NTPv3 resolves hostnames into IPv4 addresses at configuration (when the command is parsed). Then, only the resolved IPv4 address is kept in memory and stored in NVRAM during NVGEN. The hostname that is given by the user is lost.



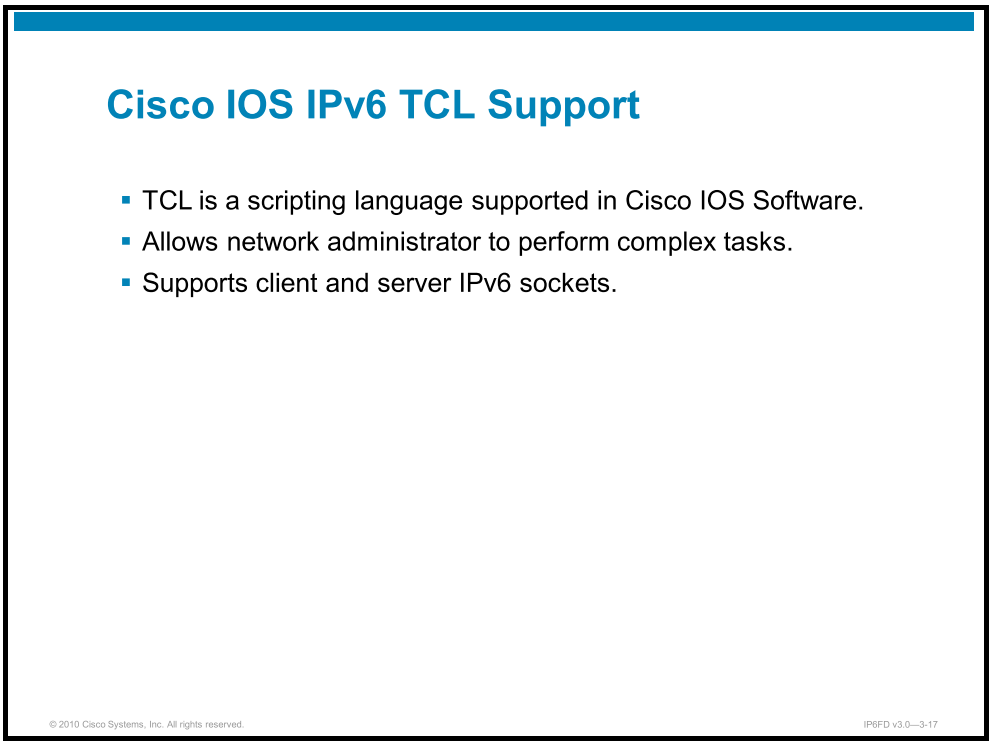
Cisco has long supported IP-MIB and IP-FORWARD-MIB in IPv4. CISCO-IETF-IP-MIB and CISCO-IETF-IP-FORWARDING-MIB are IPv6 MIBs that are defined as being protocol-independent, but are implemented only for IPv6 objects and tables. In Cisco IOS Release 12.2(33)SRC, IP-MIB and IP-FORWARD-MIB were updated to 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 only supports IPv6 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.



The Cisco IOS system message logging (syslog) process in IPv6 allows users to log syslog messages to external syslog servers and hosts with IPv6 addresses. This implementation allows the user to specify an IPv4-based logging host (syslog server) by providing the IP address of the host in IPv4 format (for example, 192.168.0.0). The user can also specify IPv6-based logging by specifying an **ipv6** parameter that is followed by the host IP address in IPv6 format (for example, ipv6 2001:0DB8:A00:1::1/64).



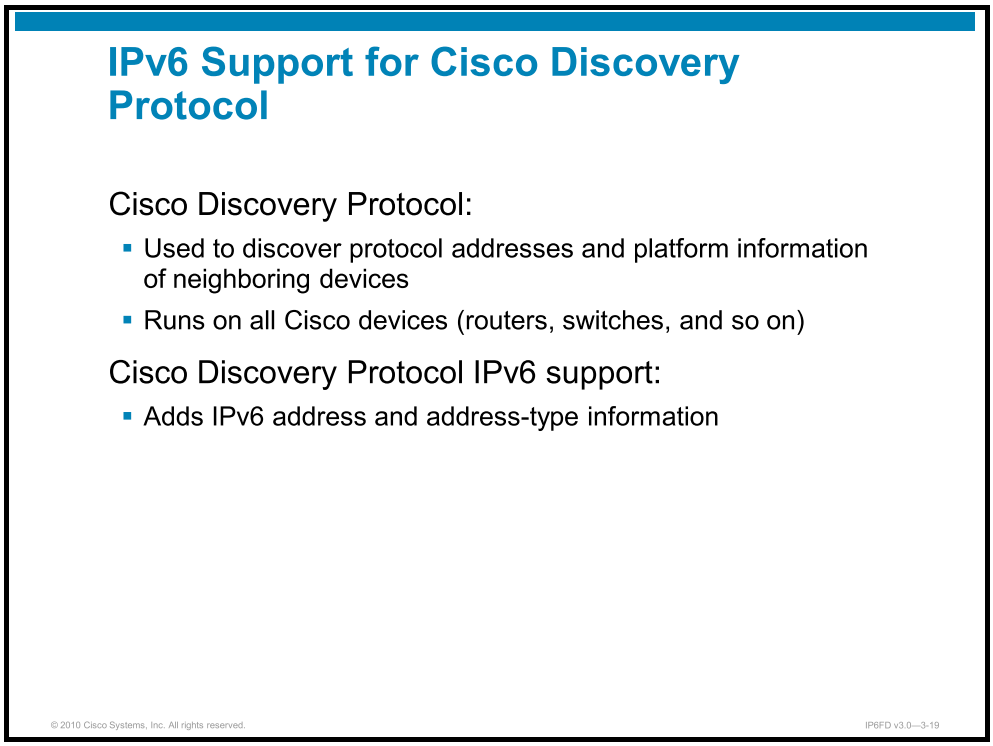
The Cisco IOS Scripting with TCL feature provides the ability to run Tcl version 8.3.4 commands from the Cisco IOS command-line interface (CLI).

Several methods have been developed for creating and running Tcl scripts within Cisco IOS Software. A Tcl shell can be enabled, and Tcl commands can be entered line by line. After the Tcl commands are entered, they are sent to a Tcl interpreter. If the commands are recognized as valid Tcl commands, the commands are executed and the results are sent to the tty device. If a command is not a recognized Tcl command, it is sent to the Cisco IOS CLI parser. If the command is not a Tcl or Cisco IOS command, two error messages are displayed. A predefined Tcl script can be created outside of Cisco IOS Software, transferred to flash or disk memory, and run within Cisco IOS Software. It is also possible to create a Tcl script and precompile the code before running it under Cisco IOS Software.

Sockets that are created using the **socket** command support both IPv4 and IPv6. UDP sockets, if supported, can be opened using **udp open** with an **–ipv6** parameter.

IPv6 Support for Cisco Discovery Protocol

This topic describes IPv6 support for Cisco Discovery Protocol on Cisco devices.



Cisco Discovery Protocol is commonly used to discover the protocol addresses of neighboring devices and the platform of those devices. Cisco Discovery Protocol is protocol-independent and supports IPv6 address and address-type information.

Cisco IOS Cisco Discovery Protocol show Commands

These commands display the IPv6 address of the neighbor:

```
router#
show cdp entry {* | device-name[*]} [version] [protocol]
show cdp neighbors [type number] [detail]
```


The **show cdp entry** and **show cdp neighbors** commands display the IPv6 address configured on the neighbor device.

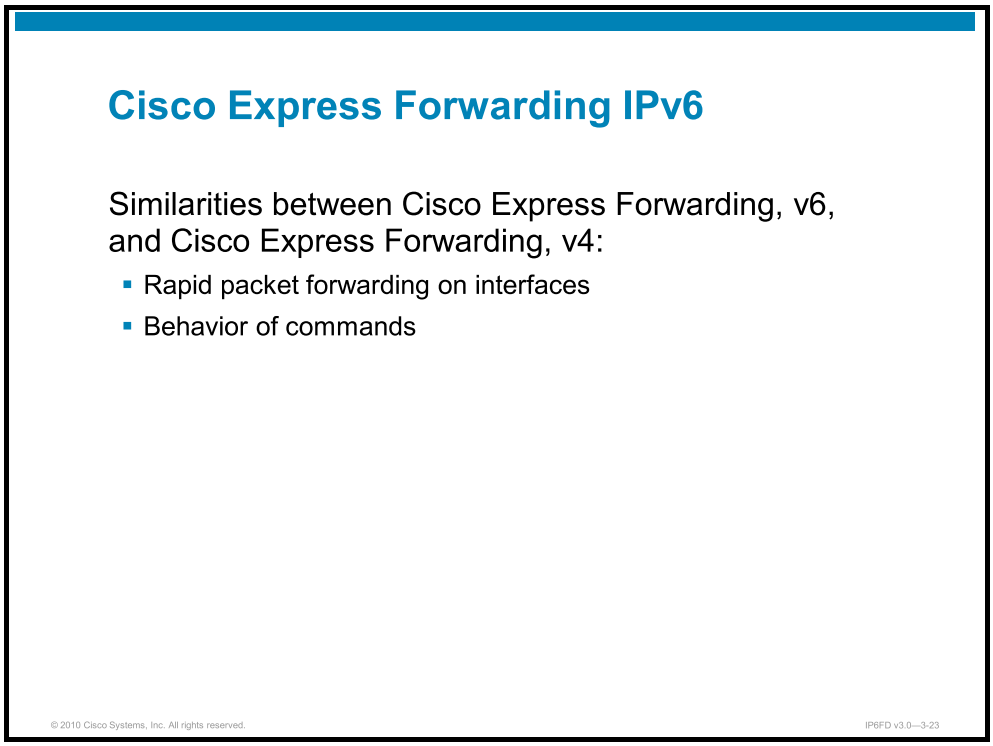
Cisco IOS Cisco Discovery Protocol show Command Example

```
routerB# show cdp neighbors detail
-------------------------
Device ID: routerC
Entry address(es):
  IPv6 address: FE80::208:A3FF:FEAE:3B81  (link-local)
  IPv6 address: FEC0::2  (site-local)
Platform: cisco 2621,  Capabilities: Router Switch
Interface: FastEthernet0/1,  Port ID (outgoing port):
FastEthernet0/1
Holdtime : 169 sec
...
```


This example demonstrates the output of the **show cdp neighbors detail** command. Note that the neighbor device, routerC, has two IPv6 addresses configured on its interface.

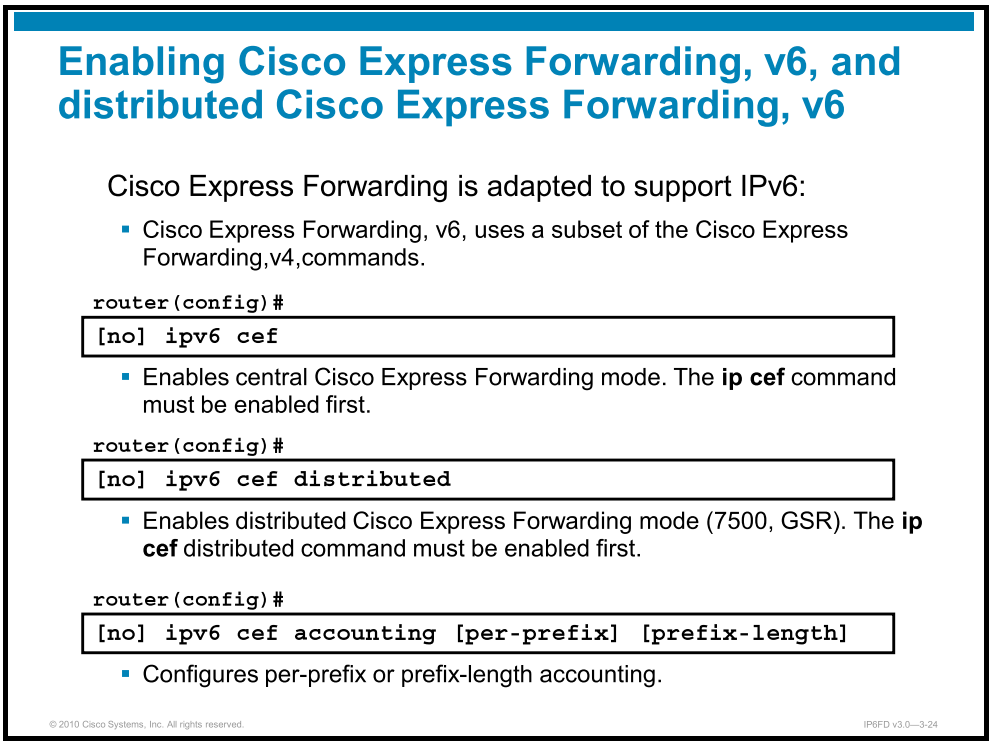
Cisco Express Forwarding IPv6

This topic describes Cisco Express Forwarding in IPv6.



Cisco Express Forwarding is a Layer 3 switching technology that is designed for routers. It uses a method that optimizes route lookups to achieve very fast traffic forwarding. Cisco Express Forwarding uses two tables to store the information that is needed for routing: the Forwarding Information Base (FIB) and the adjacency table.

The behavior of Cisco Express Forwarding version 6 is the same as that of Cisco Express Forwarding version 4. There are new configuration commands for Cisco Express Forwarding version 6 and common commands for both Cisco Express Forwarding, version 6, and Cisco Express Forwarding, version 4.



There are two Cisco Express Forwarding modes:

- **Central Cisco Express Forwarding mode:** The route processor manages Cisco Express Forwarding and adjacency tables. Cisco Express Forwarding mode is supported on Cisco Router Series 1700 to 7500.
- **Distributed Cisco Express Forwarding mode**: Distributed Cisco Express Forwarding is used on Versatile Interface Processor (VIP) and gigabit switch router (GSR) line cards. It performs the express forwarding of packets between port adapters and uses interprocess communication (IPC) to synchronize the Cisco Express Forwarding FIB and adjacency tables between router processor and line cards. This mode is supported on Cisco Router Series 7500 and 12000(GSR).

The **ipv6 cef** command enables the central Cisco Express Forwarding, version 6, mode. However, IPv4 Cisco Express Forwarding must first be enabled using the **ip cef** command. Similarly, IPv4 distributed Cisco Express Forwarding must be enabled before distributed Cisco Express Forwarding, version 6.

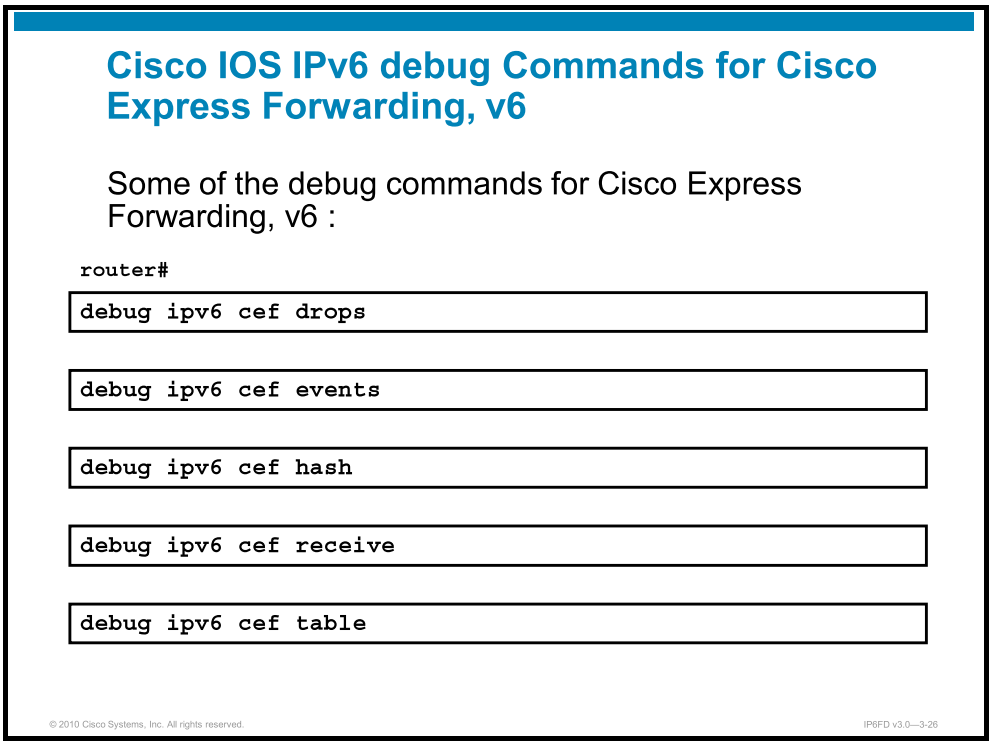
Cisco IOS show Commands for Cisco Express Forwarding, v6, and distributed Cisco Express Forwarding, v4

Some of the **show** commands available for both Cisco Express Forwarding, v6, and Cisco Express Forwarding, v4:

```
router#
show cef drop

show cef interface [detail] [statistics] interface

show cef linecard [detail] [internal] slot

show cef not-cef-switched
```


The table shows common **show** commands for distributed Cisco Express Forwarding, version 4, and Cisco Express Forwarding, version 6.

Command	Description
`show cef drop`	Shows counters of IPv6 and IPv4 packets dropped
`show cef interface [detail] [statistics] interface`	Shows CEF interface status and configuration
`show cef linecard [detail] [internal] slot`	Shows Cisco Express Forwarding information that is related to line cards
`show cef not-cef-switched`	Shows counters of IPv6 and IPv4 packets that are passed on to the next switching layer

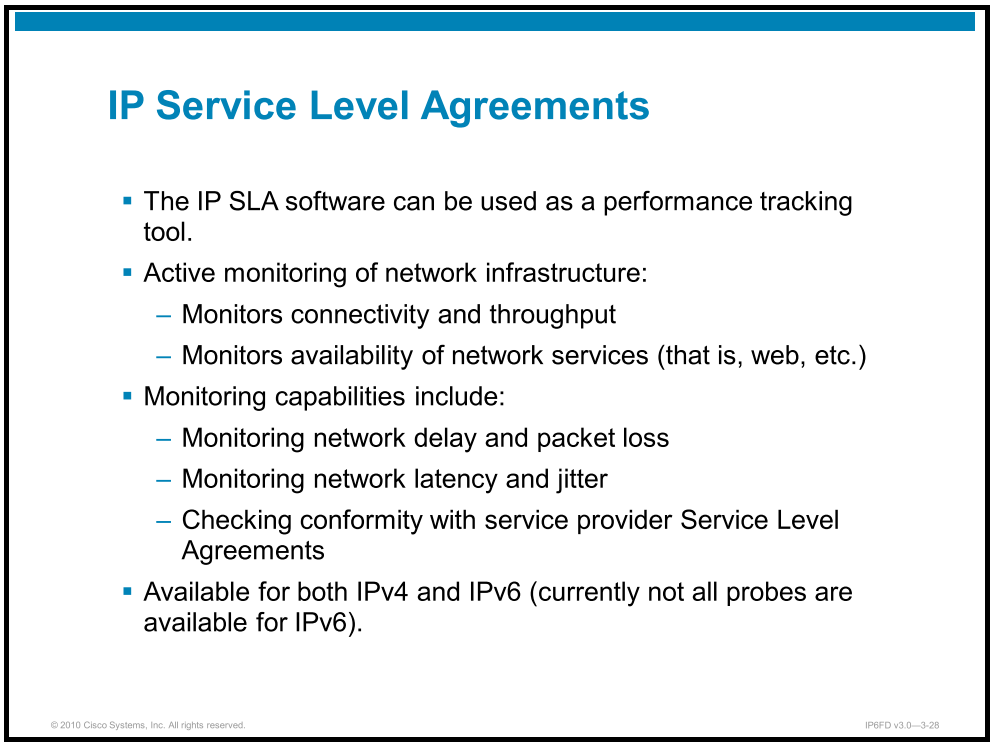


Some of the **debug** commands that are used for Cisco Express Forwarding, version 6, are shown in the table.

Command	Description
`debug ipv6 cef drops`	Debugs packets that are dropped by IPv6 Cisco Express Forwarding switching
`debug ipv6 cef events`	Debugs control plane events for IPv6 Cisco Express Forwarding
`debug ipv6 cef hash`	Debugs load-balancing hash setup events for Cisco Express Forwarding, version 6
`debug ipv6 cef receive`	Debugs packets that are passed to IPv6 process-level switching
`debug ipv6 cef table`	Debugs IPv6 Cisco Express Forwarding table modification events

IP Service Level Agreements

This topic describes the Cisco IOS IP SLA functionality when using IPv6.



Cisco IOS IP SLAs is a part of the Cisco IOS Software, which allows customers to analyze IP service levels for IP applications and services, to increase productivity and to reduce the frequency of network outages.

Cisco IOS IP SLA uses active traffic monitoring – the generation of traffic in a continuous, reliable, and predictable manner – to measure network performance. Using Cisco IOS IP SLA, customers can verify service levels, and verify internal and outsourced service level agreements, and understand network performance.

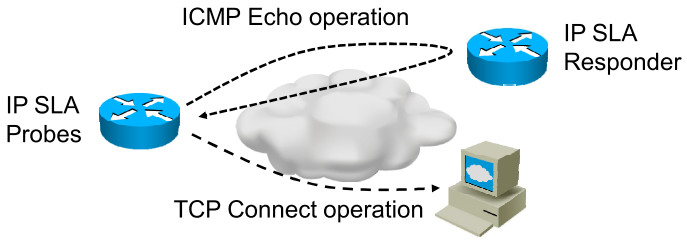
Cisco IOS IP SLAs can perform network assessments, verify quality of service (QoS), ease the deployment of new services, and assist administrators with network troubleshooting.

Cisco IOS IP SLAs can be accessed using the Cisco IOS command-line interface or SNMP through the Cisco Round-Trip Time Monitor (RTTMON) and Syslog management information bases (MIBs).

Note Cisco IOS IP SLAs originated from the technology previously known as Service Assurance Agent (SAA), or Response Time Reporter (RTR).

IP SLA Key Components

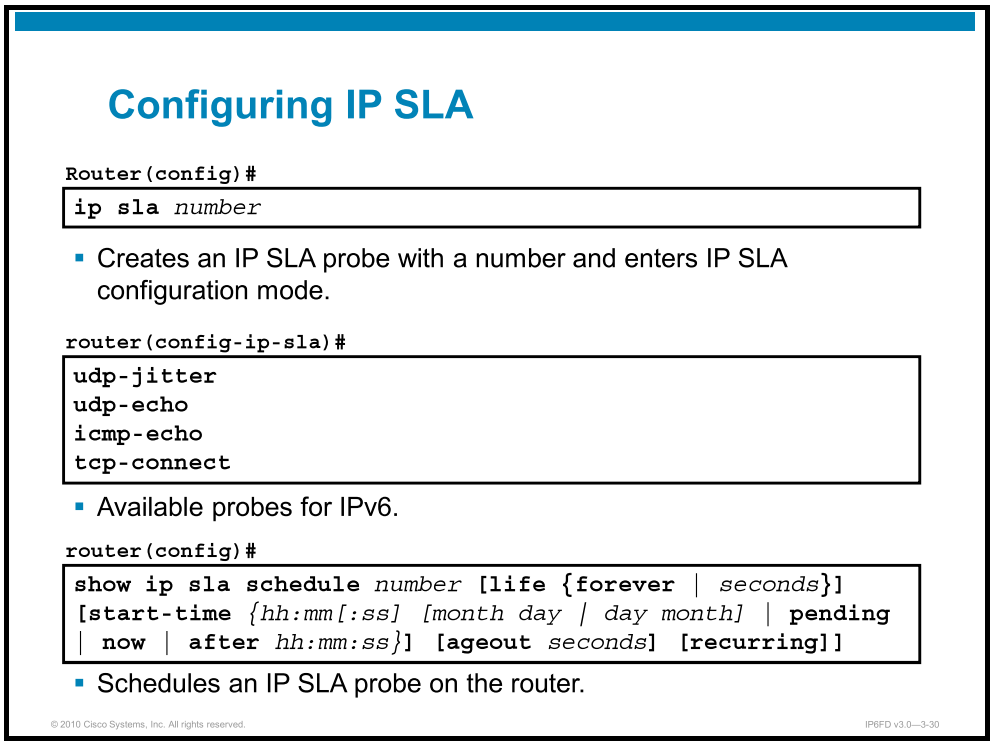
- The IP SLA architecture consists of an IP SLA Source and an IP SLA Target.
- A probe is configured on the source, checking connectivity from the source to the target.
- Target device can be a router with an IP SLA responder, or an IPv6 endpoint.



Cisco IOS IP SLA sends data across the network to measure performance between multiple network locations or across multiple network paths. It simulates network data and IP services, and collects network performance information in real time. The information collected includes data about response time, one-way latency, jitter (inter-packet delay variance), packet loss, voice quality scoring, network resource availability, application performance, and server response time.

Cisco IOS IP SLA performs active monitoring by generating and analyzing traffic to measure performance either between Cisco IOS devices or from a Cisco IOS device to a remote IP device such as a network application server. Measurement statistics provided by the various Cisco IOS IP SLA operations can be used for troubleshooting, for problem analysis, and for designing network topologies.

Cisco IOS IP SLA starts when the Cisco IOS IP SLA device sends a generated packet to the destination device. After the destination device receives the packet, and depending on the type of Cisco IOS IP SLA operation, the device will respond with time-stamp information for the source to make the calculation on performance metrics. A Cisco IOS IP SLA operation performs a network measurement from the source device to a destination in the network using a specific protocol such as UDP.



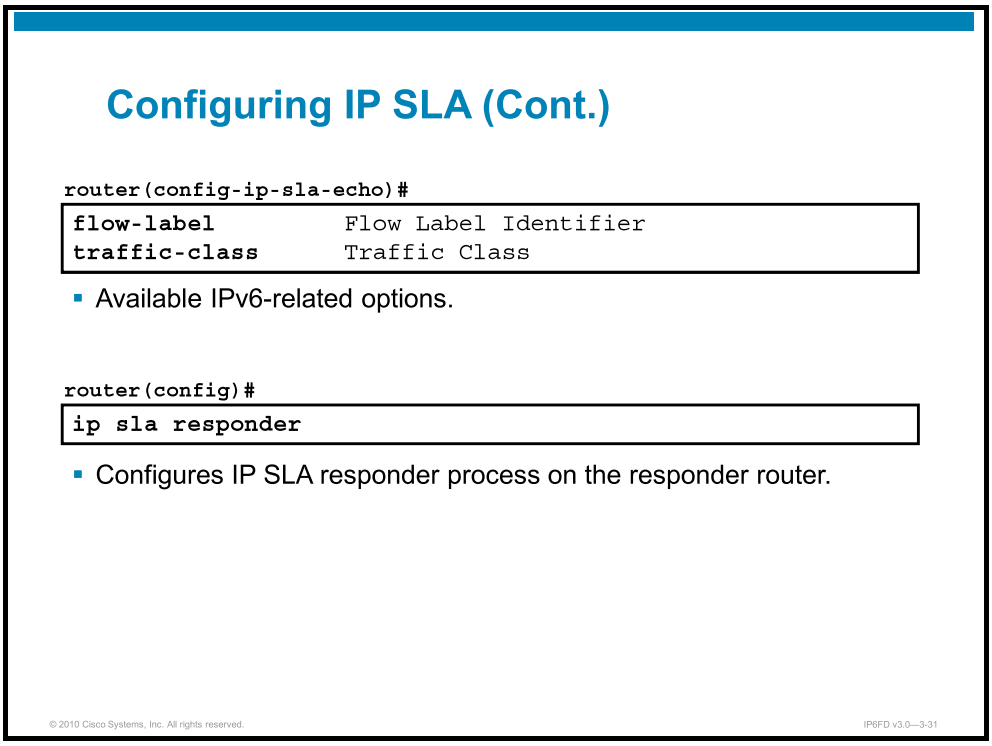
IP SLA is configured by creating an IP SLA probe on the source device, and by scheduling that probe to run at desired time.

The currently available probes of IP SLA that can be used for IPv6 are **udp-jitter, udp-echo, icmp-echo**, and **tcp-connect.** See the table for a description of functionality of these probes:

IP SLA Operation	Measurements	Key Monitoring Application
UDP Jitter	Measures round-trip delay, one-way delay, one-way jitter, one-way packet loss, and connectivity testing of networks that carry UDP traffic, such as voice. **Note**: One-way delay requires time synchronization between source and target routers.	■ Voice and data network performance ■ General IP performance **Note**: This is the most commonly used Cisco IOS IP SLAs operation.
UDP Echo	Measures round-trip delay of UDP traffic.	■ Server and IP application performance ■ Connectivity testing
ICMP Echo	Measures round-trip delay for the full path.	■ IP performance ■ Connectivity measurement
TCP Connect	Measures the time taken to connect to a target device with TCP.	■ Server and application performance

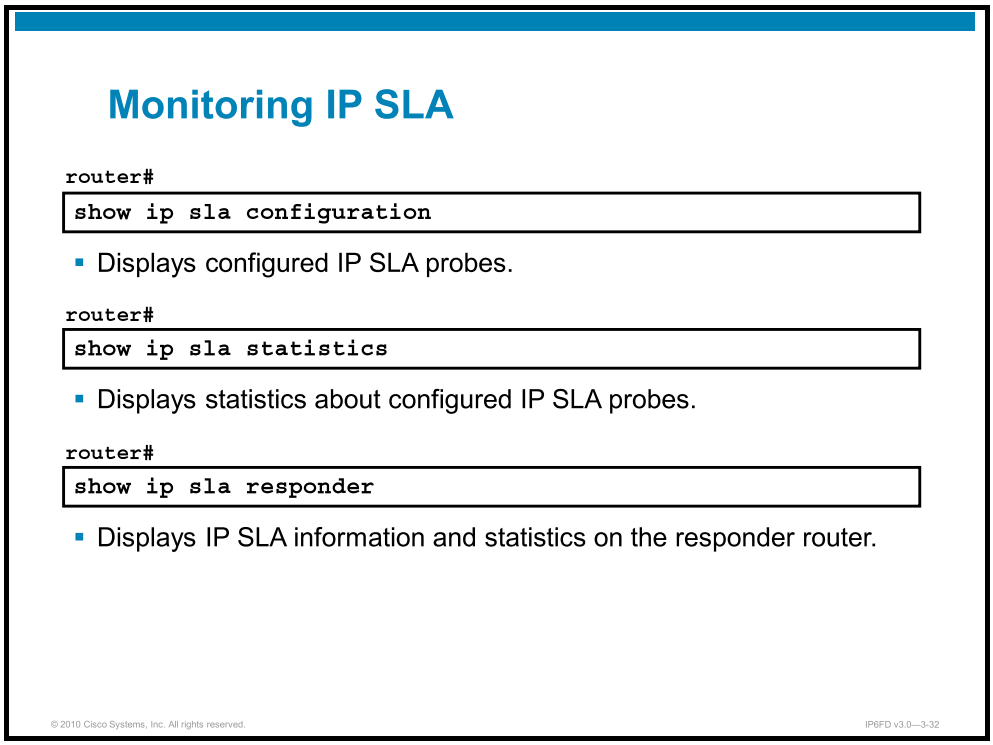
To schedule an IP SLA probe, use the **ip sla schedule** command.

Note The probe will start functioning after the IP SLA responder process has been started on the destination router.



When configuring probe options, you can set an IPv6 flow label or an IPv6 traffic class for the packets of the probe. Flow labels allow for easy identification of IPv6 packets, and traffic class allows the probe packets to be high priority, not to be dropped during congestion.

On the responder router, only one command is necessary to enable IP SLA responder.

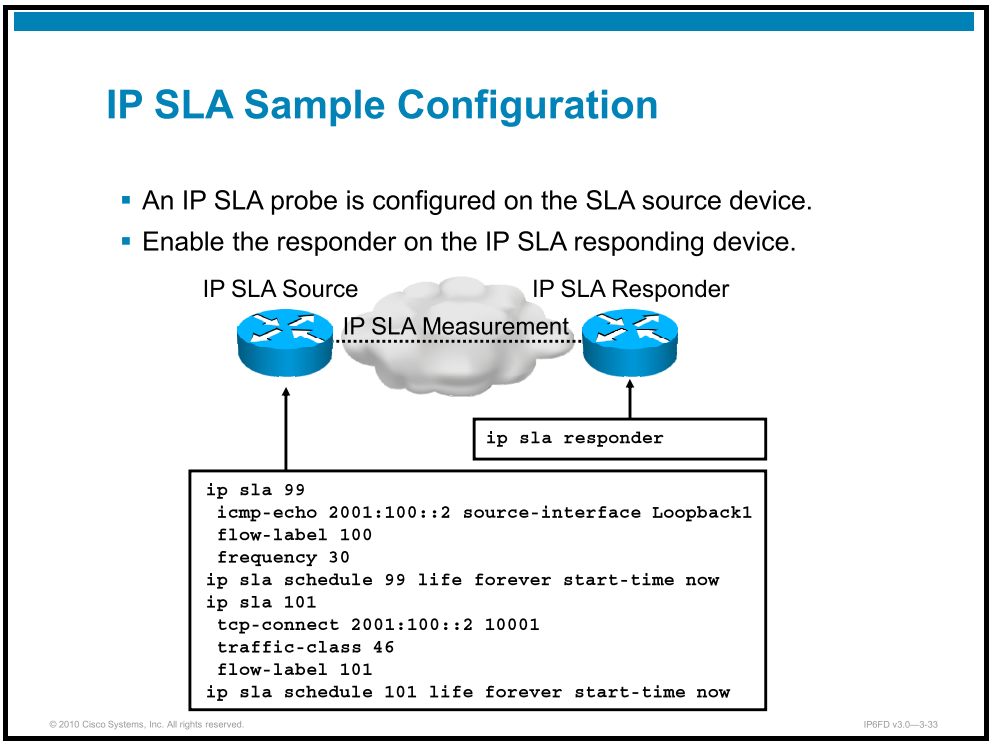


The command **show ip sla configuration** displays configuration values for all IP SLA operations or for a specific operation. The output includes all defaults.

To display the current operational status and statistics of all IP SLA operations or for a specific operation, use the command **show ip sla statistics**.

On the IP SLA responder router, use the command **show ip sla responder** to display information about the IP SLA responder.

Note	On Cisco IOS versions before 12.4(4)T and 12.2(33)S, commands **show ip sla monitor statistics** and **show ip sla monitor configuration** are used instead of **show ip sla statistics** and **show ip sla configuration**. Newer versions omit the keyword **monitor**.
Note	Previously, IP SLA was known as IP SAA (Service Assurance Agent) and RTR (Response Time Reporter).



This is a sample configuration of IP SLA if configured using the command line. On the IP SLA source device, the probe must be configured, while on the SLA responder device, only the responder must be enabled.

When IP SLA sends the packets, it uses a timestamp to determine when the packet has been sent and how much time the packet needed to travel across the networks. Based on this information, SLA statistics can be calculated.

The example shows **icmp-echo** and **tcp-connect** probes configured on the IP SLA source device. The target device in this case can be either an SLA responder, or a target end device responding to ICMP echo packets, or TCP initiated connections.

Note IP SLA is most often used as in-band monitoring. IP SLA packets are intermixed with actual data traffic, sharing bandwidth with the production traffic. Thus, if production traffic suffers from packet loss, long delay or jitter, IP SLA packets will be affected, too. Such packet loss will be visible when examining the IP SLA probe results.

Sample IP SLA Output

- Use the **show ip sla statistics** command to display statistics about currently active IP SLA probes.

```
R1#show ip sla statistics
IPSLAs Latest Operation Statistics

IPSLA operation id: 99
Type of operation: icmp-echo
        Latest RTT: 4 milliseconds
Latest operation start time: *12:53:01.218 UTC Fri May 14 2010
Latest operation return code: OK
Number of successes: 40
Number of failures: 0
Operation time to live: Forever


IPSLA operation id: 101
Type of operation: tcp-connect
        Latest RTT: 4 milliseconds
Latest operation start time: *12:53:11.750 UTC Fri May 14 2010
Latest operation return code: OK
Number of successes: 12
Number of failures: 0
Operation time to live: Forever
```


The **show ip sla statistics** displays you the statistics values about all (or a specific) IP SLA probes running on the router. Here you can read the delay, packet loss values, etc. All this data is also obtainable from the MIB, using management software that correctly interprets the data received by the Cisco RTT-MON MIB.

The example shows a printout with two probes running on the router, the IP SLA **icmp-echo** and **tcp-connect** probes. The output corresponds to the following configuration:

```
ip sla 99
 icmp-echo 2001:100::2 source-interface Loopback1
 flow-label 100
 frequency 30
ip sla schedule 99 life forever start-time now
ip sla 101
 tcp-connect 2001:100::2 10001
 traffic-class 46
 flow-label 101
ip sla schedule 101 life forever start-time now
```

Sample IP SLA Output (Cont.)

- Use the **debug ip sla trace** command on the responder.
- Relevant **debug ip sla** commands: **error, trace**.

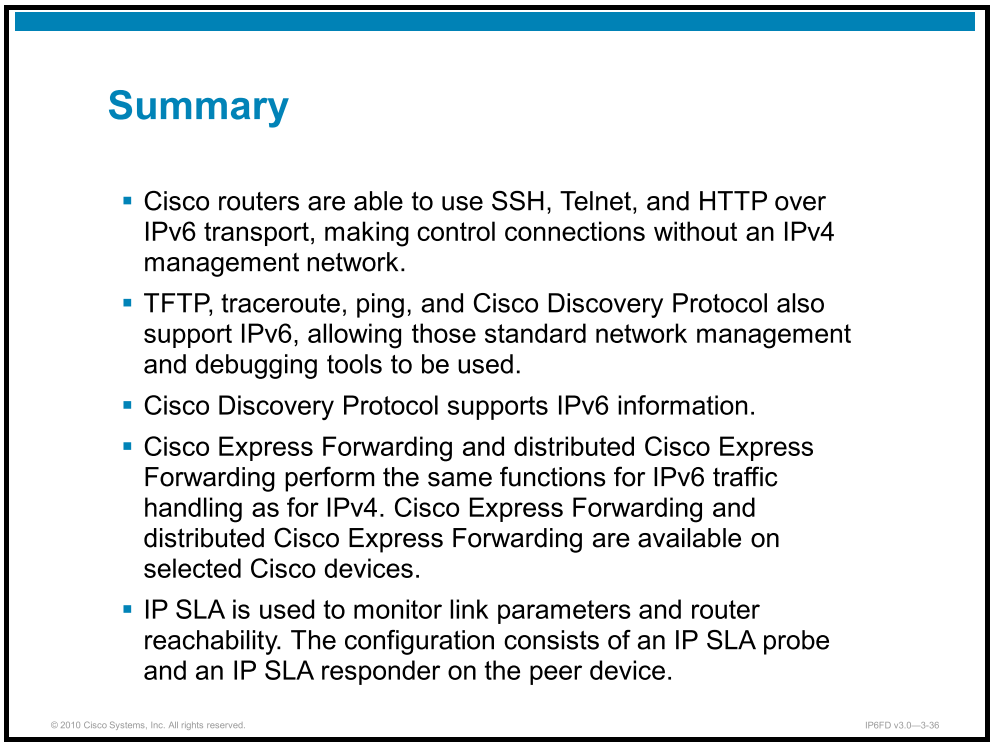
```
R1#debug ip sla trace
*May 14 13:12:13.490:  Ver: 1 ID: 31 Len: 68
*May 14 13:12:13.490: cmd: command: RTT CMD TCPV6 CONN ENABLE, ip:
2001:100::2, port: 10001, duration: 5000
*May 14 13:12:13.490: cmd: command: RTT CMD AUTH
*May 14 13:12:13.494: IP SLAs hash insert : 2001:100::2 10001
index=40
*May 14 13:12:13.494: IP SLAs RESP: starting expTimer and hashTimer,
duration = 5000
*May 14 13:12:13.494: rtt process v6 ctrl msg: Remote client
address: 2001:1::C800:3AFF:FEB6:8, Local Address: 2001:100::2,
length 68
*May 14 13:12:13.506: cleaning up port (2001:100::2,10001)
*May 14 13:12:13.506: IP SLAs hash remove: 2001:100::2 10001
```


These are some **debug** commands used for IP SLA:

- **error** Output IP SLA Error Messages
- **ethernet-monitor** Output of IP SLAs Auto Ethernet messages
- **mpls-lsp-monitor** Output of IP SLAs MPLSLM messages
- **trace** Output IP SLAs Trace Messages

Summary

This topic summarizes the primary points that are discussed in this lesson.



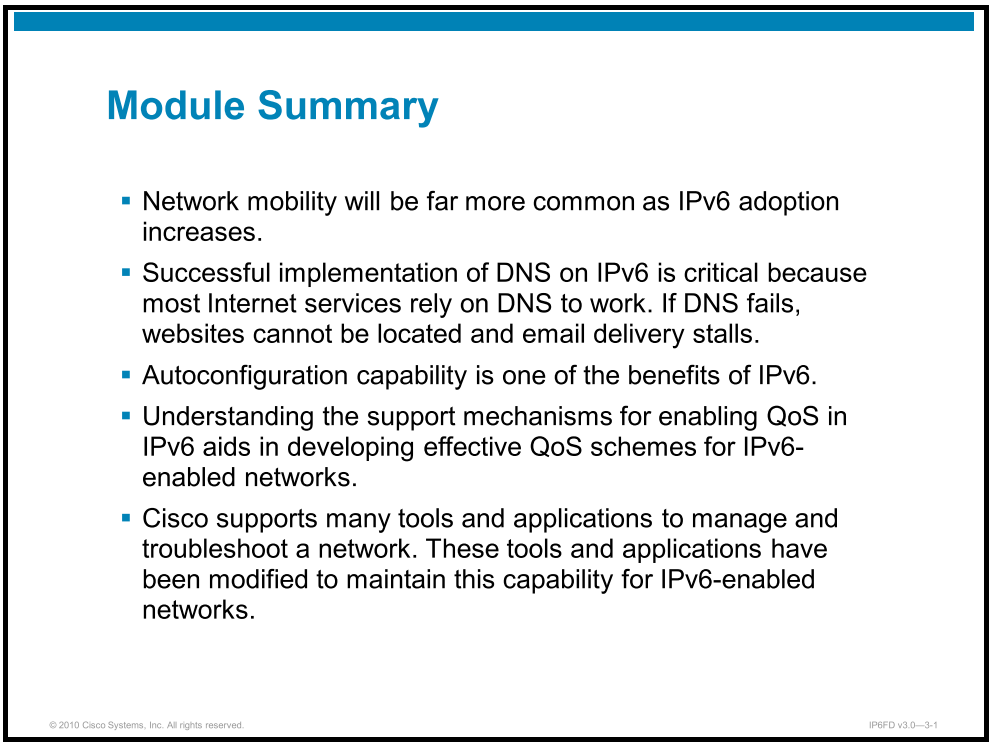
References

For additional information, refer to these resources:

- *Implementing IPv6 for Network Management* at http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-mng_apps.html#wp1055475
- *Cisco IOS IP SLAs Features Roadmap* at http://www.cisco.com/en/US/docs/ios/ipsla/configuration/guide/sla_roadmap.html

Module Summary

This topic summarizes the key points that were discussed in this module.



Module Self-Check

Use the questions here to review what you learned in this module. The correct answers and solutions are found in the Module Self-Check Answer Key.

Q1) MIPv4 normally requires the services of a device on the network that the mobile node is visiting (when away from home). This device has been eliminated in MIPv6. What is that device called in MIPv4? (Source: Describing IPv6 Mobility)

A) foreign router
B) visiting agent
C) foreign actor
D) foreign agent

Q2) MIPv4 mobile nodes, when exchanging traffic with a correspondent node, sometimes send packets that are sourced from the home network out through the foreign agent. What is one potential problem with this process? (Source: Describing IPv6 Mobility)

A) Based on IPv4 ingress-filtering rules, an upstream router from the foreign agent will drop the packet.
B) The correspondent node will not trust that the packet is truly from the mobile node if the packet does not appear to originate from the topologically correct location on the Internet.
C) Using IPv4 ingress filtering, the home agent will drop the mobile node packet as being topologically incorrect.
D) The correspondent node will not accept a MIPv4 packet that is not addressed to its care-of address.

Q3) What does MIPv6 “triangle routing” refer to? (Source: Describing DNS in an IPv6 Environment)

A) the optimized routing path that traffic takes after the mobile node-correspondent node path has been optimized
B) the inefficient path that traffic takes between the mobile node-home agent-foreign agent-correspondent node components
C) the inefficient path that traffic takes from the mobile node to the correspondent node over a home agent, before the path is being optimized, or for the whole duration of the session if the correspondent node does not support MIPv6
D) the inefficient path that traffic takes between the mobile node to the correspondent node over both the home agent and the foreign agent

Q4) MIPv6 supports the ability of the mobile node and the correspondent mode to optimize the path that traffic takes, which eliminates routing the packet through the home agent. Why does ingress filtering not drop these packets? (Source: Describing DNS in an IPv6 Environment)

A) Packets are tunneled from the mobile node to the correspondent node, IPv6-in-IPv6, where the outer IPv6 tunnel packet has a topologically correct address.
B) Packets are sent from the CoA, which is topologically correct. The correspondent node accepts the packets as being from the mobile node because of information that is passed to the correspondent mode in the mobile node-correspondent node bootstrapping phase 1 process.
C) A requirement of MIPv6 deployments is that all routers in the path between the mobile node and correspondent node disable ingress filtering, so that packets marked as MIPv6 are never dropped for having a topologically incorrect source address.
D) Packets are sent from the mobile node to the correspondent node using the Home Address Option (HAO), which is placed in the Destination Options extension header, so that the source address of the packet is topologically correct. It is the mobile node CoA.

Q5) What does using Network Mobility (NEMO) allow? (Source: Understanding DHCPv6 Operations)

A) a single /64 subnet to use the service of a mobile router, even though the nodes on the /64 remain stationary
B) a collection of networks, or a single subnet, to be mobile as a group behind a mobile router, in which the nodes and the mobile router move together and attach to the larger network at arbitrary locations
C) a mobile network to use IPv4 as a transport provider while the IPv6 network moves around the public Internet, or other IPv4-only network, attaching at different points and receiving a topologically correct local IPv4 address
D) mobile networks to self-organize into supernets and enable route summarization, improving the scalability for mobile networks

Q6) What are two of the characteristics of MANET networks? (Choose two.) (Source: Understanding DHCPv6 Operations)

A) low power, wireless connectivity
B) constantly changing topology
C) fixed infrastructure components
D) wired connectivity between the fixed APs and switched infrastructure

Q7) What are the header fields used for QoS that are functionally similar in IPv4 and IPv6 called? (Source: Understanding QoS Support in an IPv6 Environment)

A) Type of Service, Flow Label
B) Type of Setting, Traffic Setting
C) Type of Setting, Traffic Class
D) Type of Service, Traffic Class

Q8) The Flow Label field in IPv6 is immutable. What does “immutable” mean? (Source: Understanding QoS Support in an IPv6 Environment)

A) The flow label is selected by the source node and must be delivered intact to the destination node.
B) The flow label is selected by the source node and is eligible for rewriting by intermediate devices.
C) The flow label has no intrinsic meaning, so low values or high values of the 20-bit field convey no priority information.
D) The flow label meaning is absolute, and always takes priority over other QoS mechanisms.

Q9) How does Cisco IOS Software support Telnet for IPv6? (Source: Using Cisco IOS Software Features)

A) includes a multiprotocol implementation of Telnet, so it listens for IPv4 and IPv6 connections
B) supports IPv4 and IPv6 separately via two enabling commands (**telnet enable** and **ipv6 telnet enable**)
C) supports IPv4 and IPv6 via a single application, but does allow you to selectively disallow IPv6 Telnet connections via the **no ipv6 telnet** command
D) listens for IPv4 and IPv6 Telnet connections via the **enable telnet all** command

Q10) How does Cisco Discovery Protocol support IPv6? (Source: Using Cisco IOS Software Features)

A) Cisco Discovery Protocol runs over IPv6, in addition to IPv4, as a configuration-level setting.
B) Cisco Discovery Protocol has been ported to IPv6 only, and regardless of the protocols that are used on the data plane, Cisco routers will use IPv6 on the control plane for management functions such as Cisco Discovery Protocol.
C) Cisco Discovery Protocol provides IPv6 information, in addition to IPv4 information, when reporting on Cisco device neighbors that are discovered via Cisco Discovery Protocol.
D) Cisco Discovery Protocol maintains the IPv6 neighbor cache data structures and uses them to find Cisco Discovery Protocol neighbors.

Q11) What does Cisco Express Forwarding provide as a Cisco feature? (Source: Using Cisco IOS Software Features)

A) rapid packet forwarding for IPv4 and IPv6
B) rapid packet forwarding on Cisco 12000 Series Routers only
C) rapid packet forwarding solution for IPv6 and IPv4 FastPacket forwarding that is handled by distributed Cisco Express Forwarding
D) rapid packet forwarding in Cisco low-end routers via customer ASICS

Module Self-Check Answer Key

Q1) D

Q2) A

Q3) C

Q4) D

Q5) B

Q6) A, B

Q7) D

Q8) A

Q9) A

Q10) C

Q11) A

Module 4

IPv6-Enabled Routing Protocols

Overview

Routing protocols must support IP version 6 (IPv6) to facilitate the successful transport and operations of IPv6-generated traffic. This module describes the changes that you must make to routing protocols to accommodate IPv6 and how you must change your network environment to support IPv6 operations.

Module Objectives

Upon completing this module, you will be able to understand the updates to IP version 4 (IPv4) routing protocols needed to support IPv6 topologies. This ability includes being able to meet these objectives:

- Describe RIPng, including operations, configurations, and commands
- Describe OSPFv3 and the IPv6-capable version of the OSPF routing protocol, including its operations, configuration, and commands
- Describe IS-IS protocol, including concepts, operations, configurations, and commands
- Describe Cisco EIGRP, including operations, configuration, and commands
- Describe MP-BGP, including operations, configurations, and commands

Lesson 1

Routing with RIPng

Overview

Routing Information Protocol (RIP) is an industry stalwart, serving the routing protocol needs of smaller networks. Although the use of RIP has diminished over the years, the protocol has been updated to support IPv6 networks. This lesson describes Routing Information Protocol next generation (RIPng)—the IP version 6 (IPv6)-capable version of RIP—including its operations, configuration, and commands.

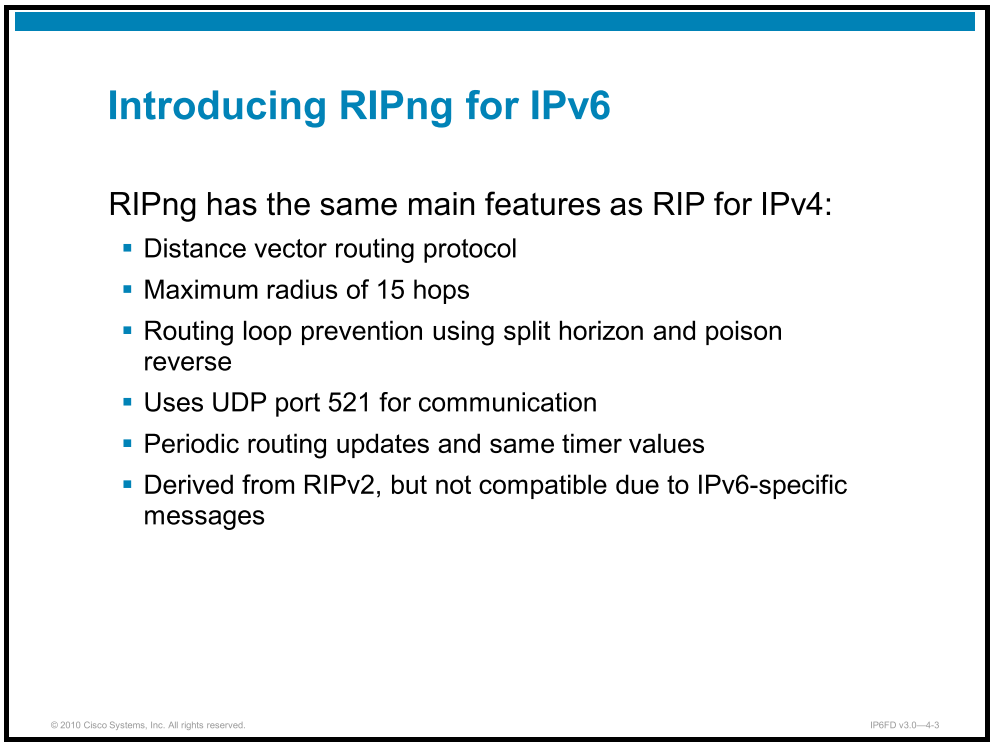
Objectives

Upon completing this lesson, you will be able to describe RIPng and configure it on Cisco routers. This ability includes being able to meet these objectives:

- Describe how RIPng is supported in IPv6
- Describe the enhancements made to RIPng to support IPv6
- Configure RIPng on Cisco routers

Introducing RIPng for IPv6

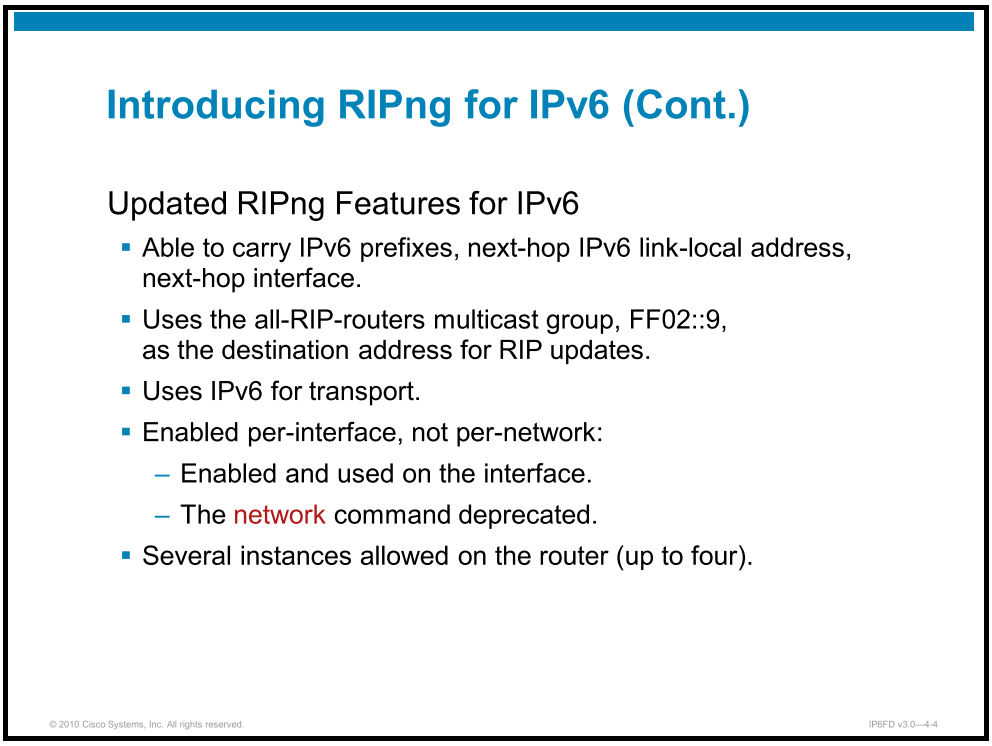
This topic describes how RIPng is supported in IPv6.



The core features of RIPng are the same as the features in Routing Information Protocol version 2 (RIPv2). RIPng remains a distance vector routing protocol with a maximum radius of 15 hops, and it uses split horizon and poison reverse to prevent routing loops in the RIPng environment.

RIPng uses native IP version 6 (IPv6) packets for transporting routing updates, using a well-known multicast address and a User Datagram Protocol (UDP) session. It is not directly compatible with RIPv2, because it uses a different update message format to be able to exchange IPv6 routes.

Note RIPng for IPv6 is defined in RFC 2080.



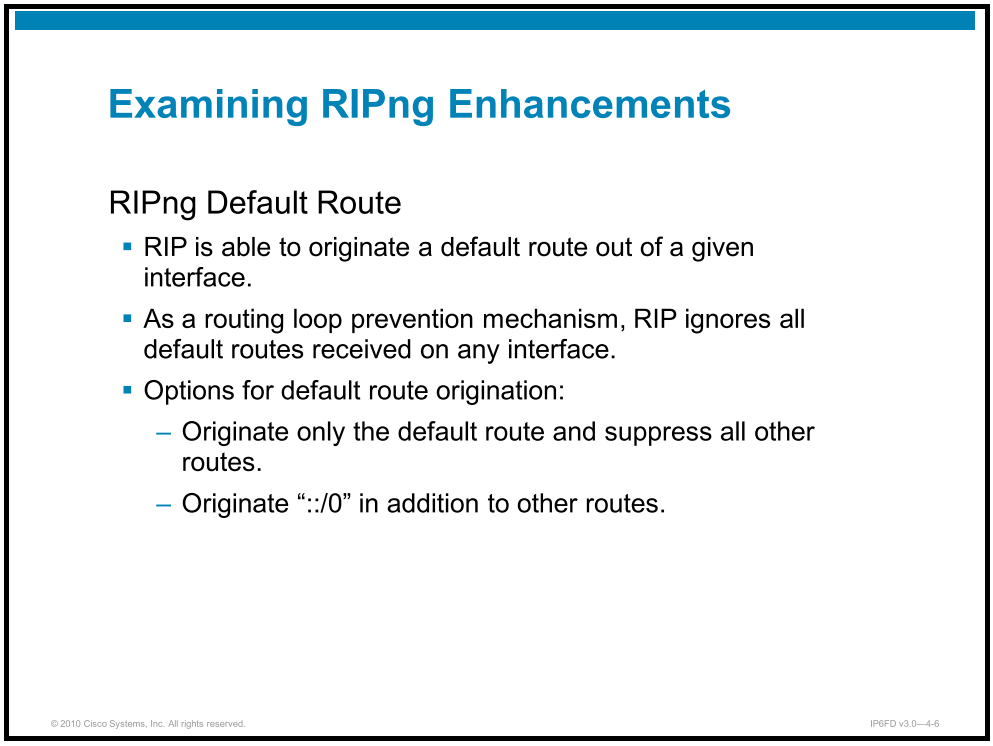
RIPng updates RIP to support IPv6 in these ways:

- IPv6 is used to transport RIPng updates.
- The IPv6 multicast address, FF02::9, is used by routers to exchange RIP updates.
- RIPng uses the link-local address of the next-hop interface in its routing table, instead of a global address.
- RIPng is enabled on a per-interface basis, rather than per-network as in RIPv2.

Tip RIPng is used mostly for labs and small businesses. The restrictions on the maximum network diameter, the simple metrics (hop count), and the length of time for convergence in any larger network make it less suitable for large production uses. Still, RIPng is a simple routing protocol for small environments and is excellent for learning about routing operations because it is simple and straightforward to configure. RIPng has been standardized for IPv6 since January 1997.

Examining RIPng Enhancements

This topic describes the enhancements made to RIPng to support IPv6, such as the ability to announce a default route, route redistribution capabilities, and multipathing.



RIPng Default Route Announcement

RIPng can announce a default route out of a given interface on the router.

RIPng includes a few enhancements to accommodate IPv6 networks. However, the main principles of functioning remain identical to RIPv2 for IP version 4 (IPv4) networks. These include route filtering, redistribution, default route origination, and so on.

RIP is most often used to originate a default route on the network edge device. For small (branch) networks, such a setup is adequate. In addition to the default route, nothing else is needed, and the router can be configured to suppress all other routes and to announce only the default route.

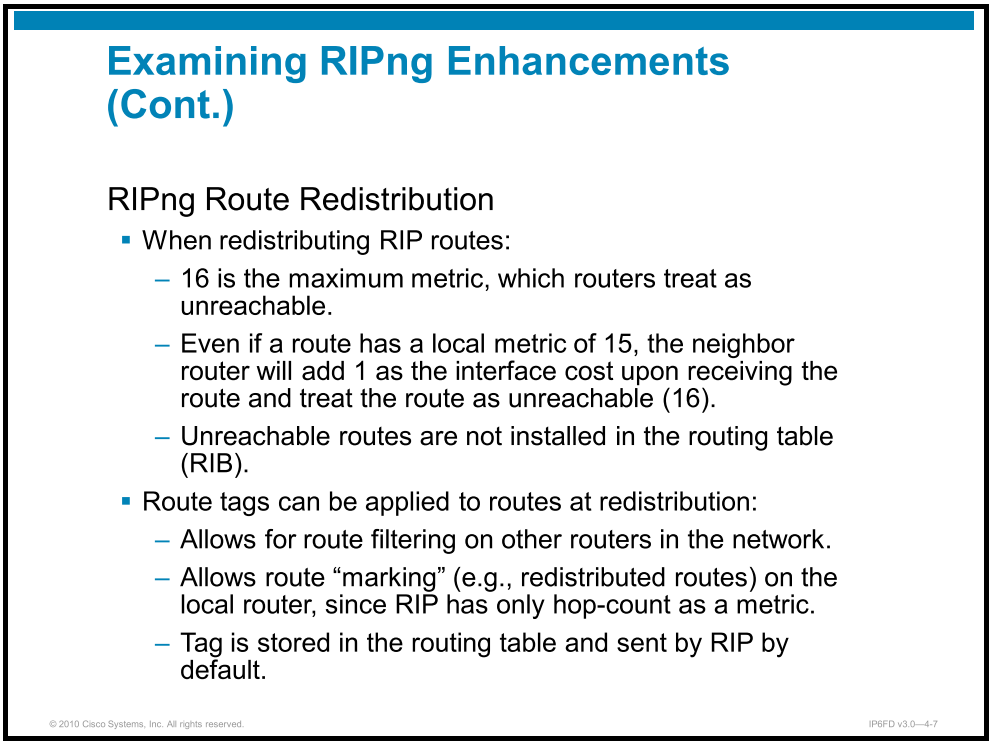
For larger branch networks, or when the branch is multihomed to two distinct network hubs, a default route can be announced from both branch routers, with more specific routes that allow branch neighbor routers to select an optimal exit point towards the correct hub.

Routing is performed on the longest match (the same as IPv4 routing), so a more specific route to a given subnet always takes precedence over the "::/0."

As a routing loop prevention mechanism, RIPng does not originate a default route on those interfaces where it receives a default route from other routers.

RIPng Route Redistribution Capabilities

RIPng is able to redistribute routing information with other IPv6 routing protocols.



Route redistribution code has been modified to accommodate IPv6 routes. Route redistribution is used to inject routers from one routing protocol into another routing protocol.

Since RIP is mainly used on branch routers, the usual redistribution will be from Border Gateway Protocol (BGP) into RIP, for the routes from other sites, and from static into RIP for local routes.

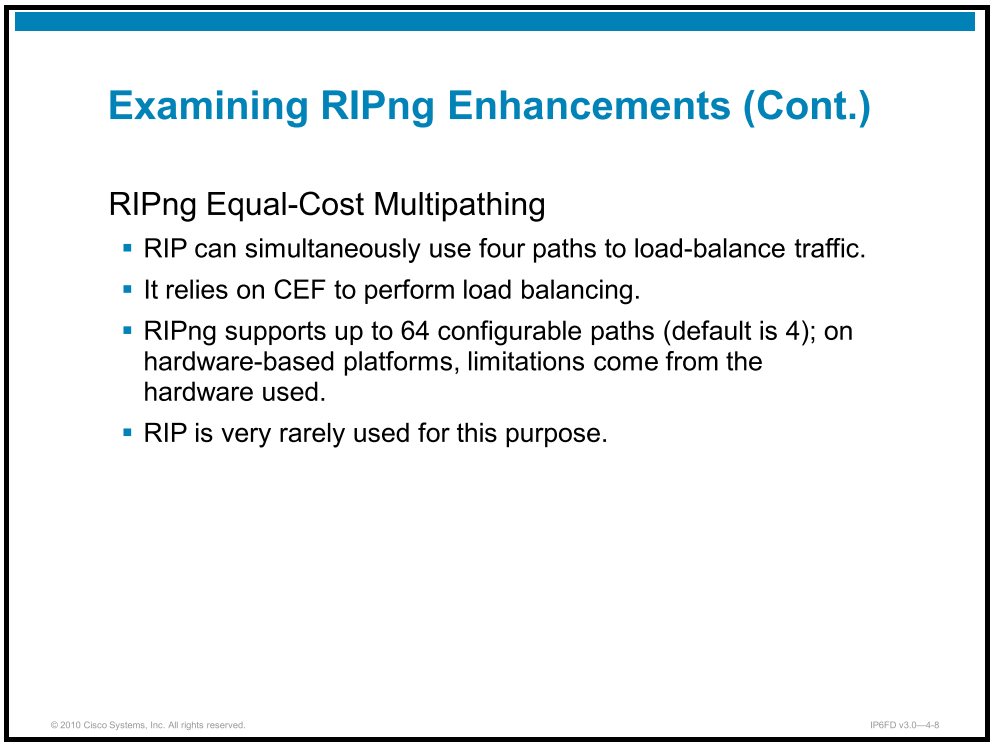
Manual metric setting is required for redistribution into RIP. By default, RIP will assign the maximum metric to the redistributed route (16), making the route unreachable. Setting the metric to 15 would install the route in the local routing table. However, neighbor routers (after adding 1 for the interface cost for the received route) would not, having a metric of 15+1=16. This means that useful metrics start from 14 and downwards.

When announcing routes, RIP does not differentiate from internal and redistributed (external) routes. To mark the routes and distinguish them one from another, we can use the route tag. A route tag can be applied to a route during redistribution, and it will be stored in the routing table. When a tag is appended to a route in the routing table, RIP will pass it along with the route. This is the default behavior.

Note There is a lesson dedicated to route redistribution later in this module.

RIP Equal-Cost Multipathing

RIPng supports load balancing across multiple paths.



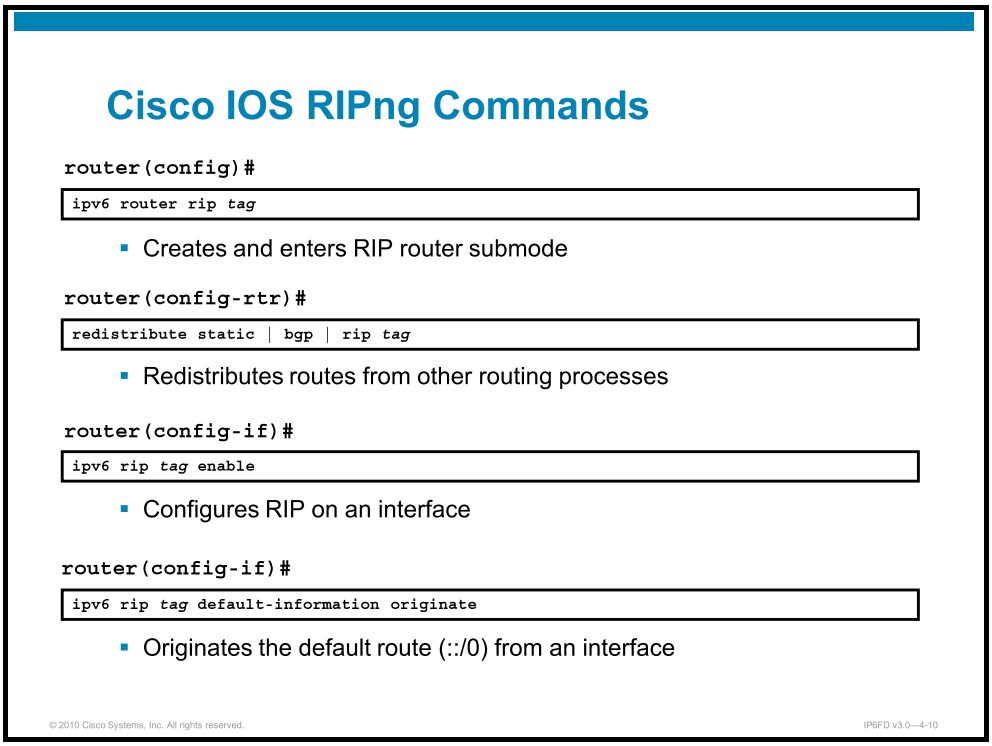
RIPng supports up to 64 paths that can be used for the load balancing of traffic. In RIP up to 4 paths are used to perform load balancing by default. The principal condition is that the cost of the route towards a given network is the same across all of these paths. This is called equal-cost multipathing, and RIPng treats such paths with equal priority.

The actual load balancing is performed on the data plane of the router; the underlying mechanism is Cisco Express Forwarding. Depending on the hardware platform, Cisco Express Forwarding might be able to load-balance as many as 16 paths of equal cost.

However, combinations of equal-cost multipathing and RIP are very rarely found in networks.

Configuring RIPng

This topic describes how to configure RIPng on Cisco routers.



The syntax of the following commonly used commands is different, compared to RIP for IPv4 counterparts:

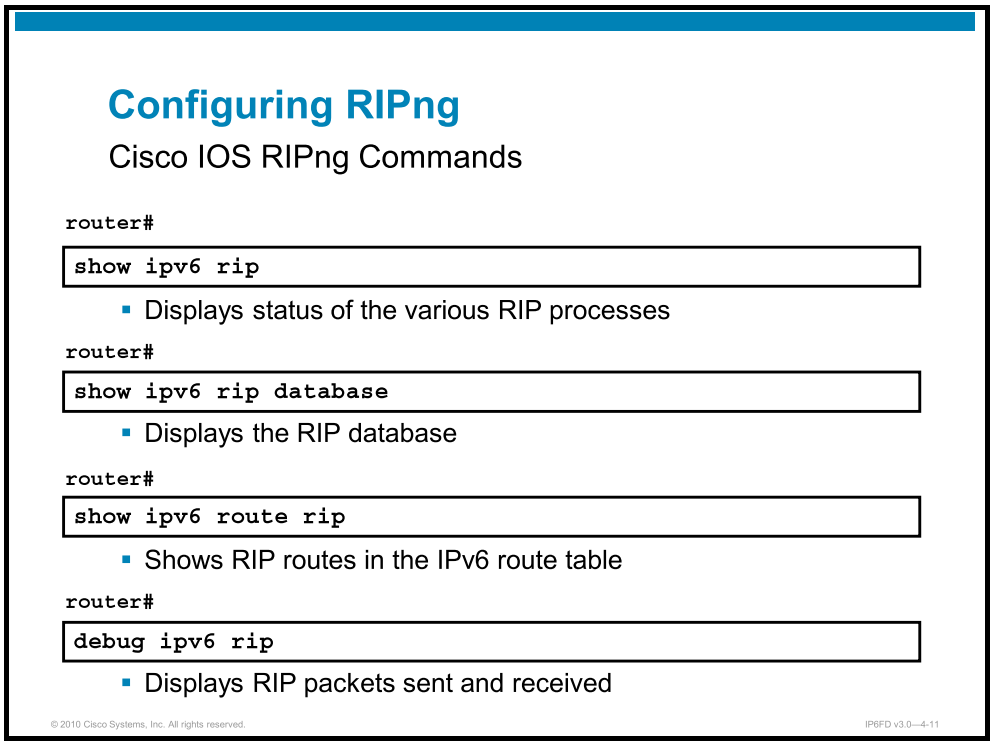
- **ipv6 router rip**
- **redistribute**
- **ipv6 rip enable**
- **ipv6 rip default-information originate**

In RIP for IPv4, you configured a routing process in the global configuration mode and added the interfaces using the **network** command. RIPng is configured similarly as Open Shortest Path First (OSPF) in interface configuration mode.

Enabling RIPng on an interface without starting the routing process in the global configuration mode first will result in a dynamically created “router rip” process in the configuration.

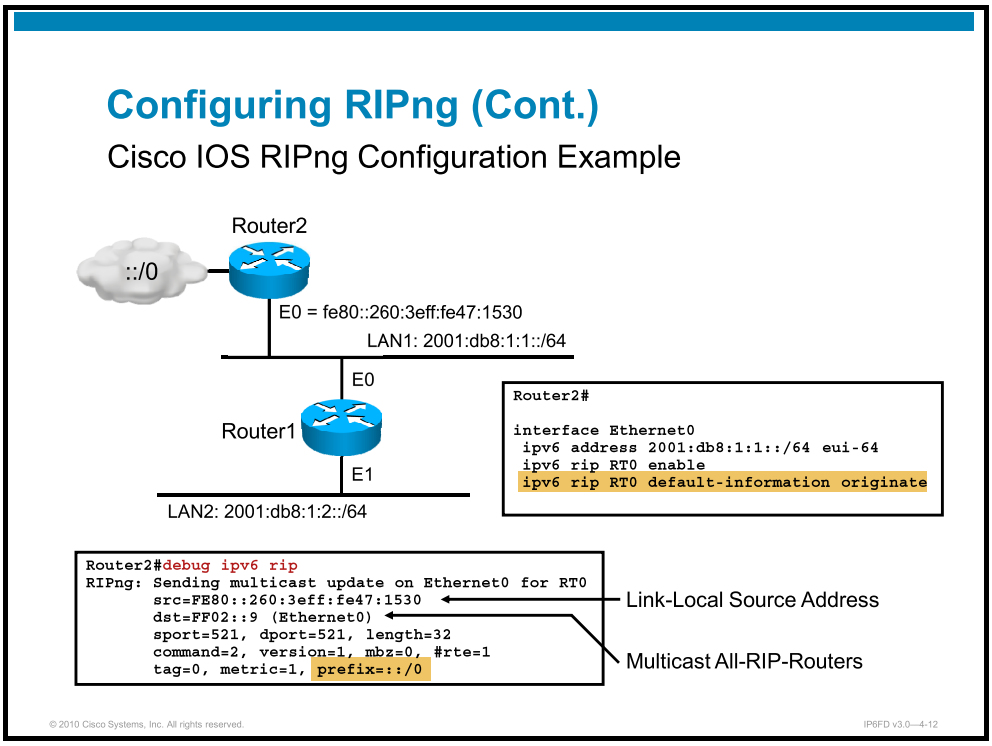
The tag for the RIPng routing process is an alphanumeric string and must be unique to the routing process. Per Cisco IOS Release 12.4T, you can configure as many as four RIPng routing processes on a router.

Note Do not confuse the routing process tag for the route tag, which is used for route redistribution.



The syntax of the following **show** and **debug** commands, which are used to verify the status of RIPng, is similar—if not identical—to the IPv4 counterpart:

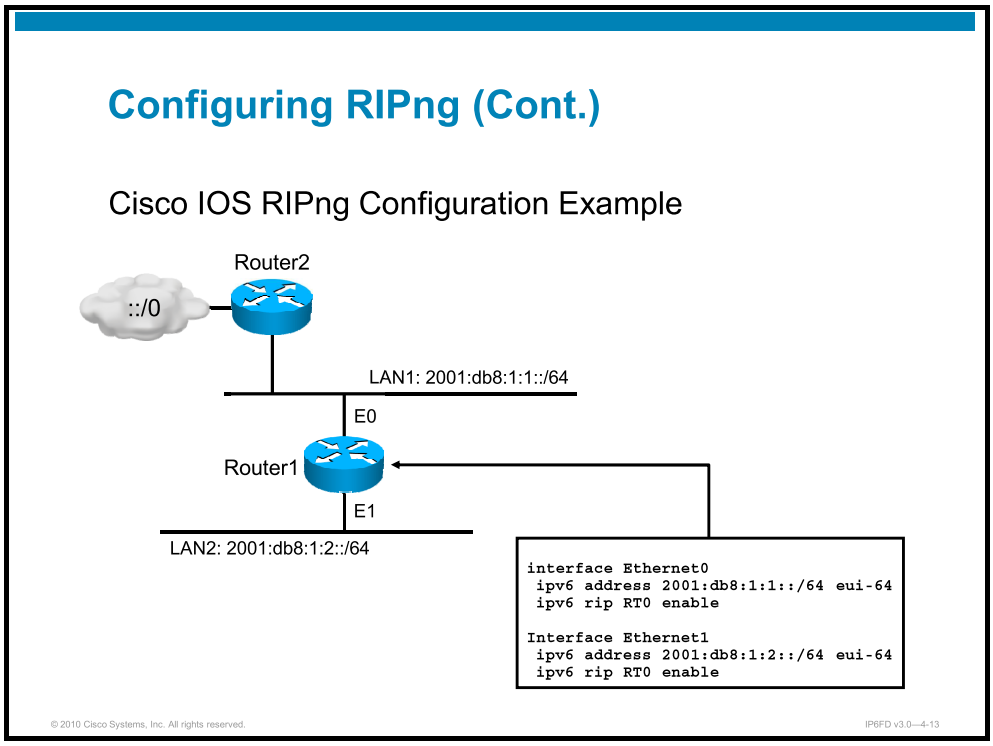
- **show ipv6 rip**
- **show ipv6 rip database**
- **show ipv6 route rip**
- **debug ipv6 rip**



The figure shows a network of two routers. The upper router, Router2, is connected to the default network. The screen text is from Router2. “RT0” is a tag to identify the RIP process. RIP is enabled on the first Ethernet interface (**ipv6 rip RT0 enable**).

There are many ways to configure default routes. In the first Ethernet interface context, **ipv6 rip default-information originate** enables the advertisement of the default route with the origin of Router2.

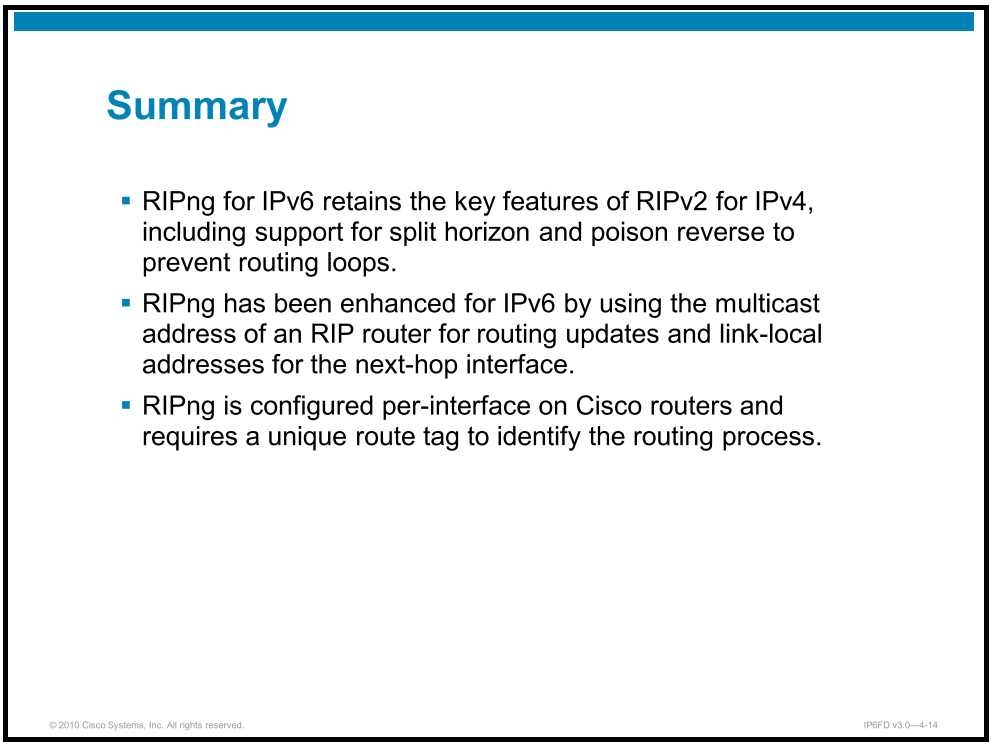
The **debug ipv6 rip** command helps with debugging. This screen output shows an RIP update being sent on the first Ethernet interface. The source address of the routing update packet is the link-local address on that network. The host ID of the link-local address is based on the extended universal identifier 64-bit format (EUI-64) representation of the data link layer MAC address. Manual address configuration for host part is also possible, but not required. The destination IPv6 address of the RIP advertisement packet is the multicast group of all-RIP-routers. The debug output shows that the router is sending a default route (prefix=::/0) with a metric of one.



The figure shows an example of a two-router portion of a larger network. The lower router is connected to two internal LANs. The screen text is from the lower router, called Router1. It shows that RIP is enabled on both Ethernet interfaces (**ipv6 rip RT0 enable**).

Summary

This topic summarizes the key points discussed in this lesson.



References

For additional information about RIPng, refer to these resources:

- *RFC 2080—RIPng for IPv6*
  http://www.faqs.org/rfcs/rfc2080.html
- *Cisco IOS IPv6 Configuration Guide, Release 12.4T: Implementing RIP for IPv6*
  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-rip_ps6441_TSD_Products_Configuration_Guide_Chapter.html

Lesson 2

Examining OSPFv3

Overview

Open Shortest Path First (OSPF) is a widely used interior gateway protocol (IGP). Upgrading the protocol to support IP version 6 (IPv6) generated a number of significant changes to how the protocol behaves. Understanding the differences between OSPF version 2 (OSPFv2) and OSPF version 3 (OSPFv3) are required for the successful deployment and operation of an IPv6 network using OSPF for routing. This lesson describes OSPFv3, the IPv6-capable version of the OSPF routing protocol, including its operations, configuration, and commands.

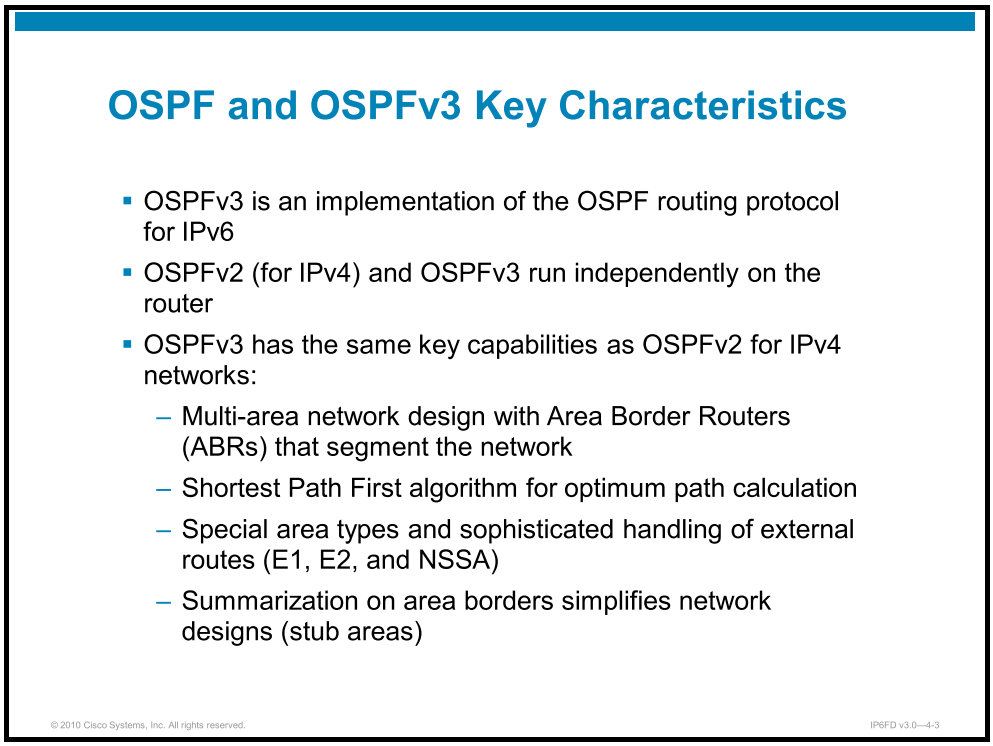
Objectives

Upon completing this lesson, you will be able to describe OSPFv3 and configure it on Cisco routers. This ability includes being able to meet these objectives:

- Describe how the OSPFv3 routing protocol is supported in IPv6
- Describe the changes and enhancements made to OSPFv3 to support IPv6
- Configure the OSPFv3 protocol on Cisco routers
- Describe OSPFv3 IPsec ESP authentication and encryption
- Describe OSPFv3 advanced functionalities

OSPFv3 Key Characteristics

This topic describes how the OSPFv3 routing protocol is supported in IPv6.



The Open Shortest Path First is a well known link-state routing protocol, suitable for large enterprise networks.

OSPF segments the network into multiple areas which communicate through area border routers. This approach allows greater scalability and relieves the routers from running route calculations for events that are not in their area. Only ABRs need to know the exact topology of all attached areas. These routers send appropriate routes as inter-area routes.

OSPF handles external routes differently than internal routes. These routes are propagated across all areas in a special update packet, and distinguished in the routing table. Special area types, such as stub areas and NSSA allow for handling external routes and summarization.

The core algorithm for best path calculation is the Shortest Path First, or Dijkstra algorithm. This algorithm is run every time when there is a topology change in the area.

Open Shortest Path First version 3 (OSPFv3) is a complete rewrite of the OSPF protocol to support IPv6. The foundation remains, for the most part, the same as in IP version 4 (IPv4) and OSPF version 2 (OSPFv2).

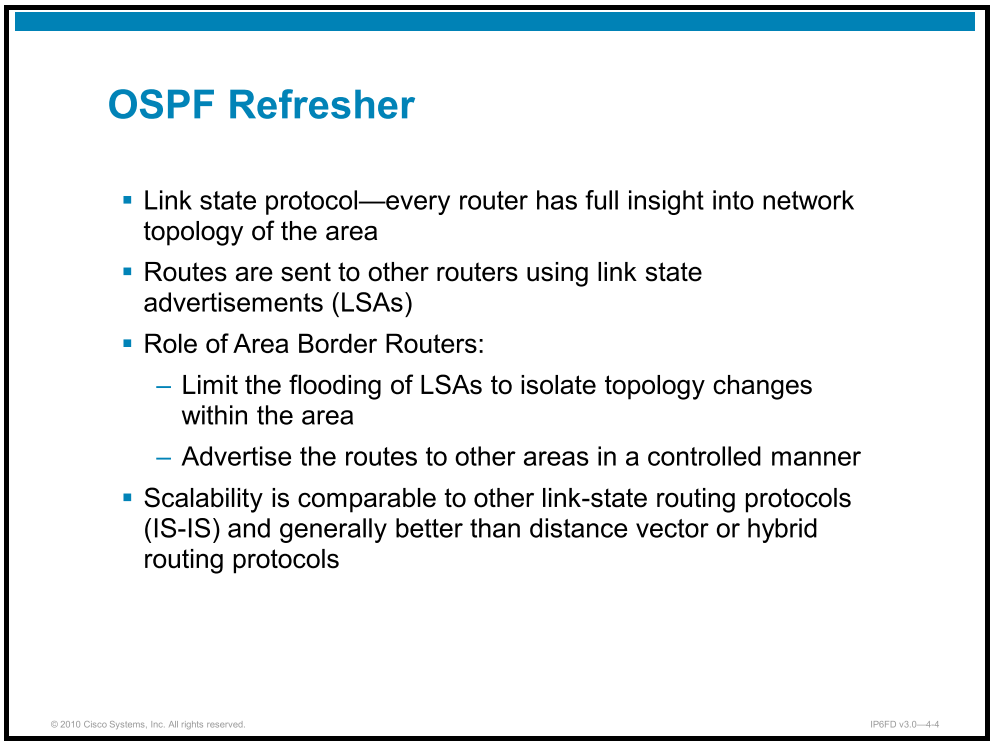
Note	OSPFv3 and OSPFv2 run independently on a router. This is a key difference to IS-IS, which can have a single process and a single topology database for both routed protocols, i.e. IPv6 and IPv4.

The OSPFv3 metric is still based on interface costing. OSPF for IP version 6 (IPv6 is currently supported in Cisco IOS Software.

The packet types and neighbor discovery mechanisms are the same in OSPFv3 as they are for OSPFv2. OSPFv3 also supports the same interface types, including broadcast, point-to-point, point-to-multipoint, nonbroadcast multiaccess, and virtual links.

Link-state advertisements (LSAs) are still flooded throughout an OSPF domain, and many of the LSA types are the same, though a few have been renamed or newly created.

Note OSPFv3 is defined in RFC 5340.



OSPF follows the same rules of operation whether it is run for IPv4 (OSPF version 2) or IPv6 (OSPF version 3).

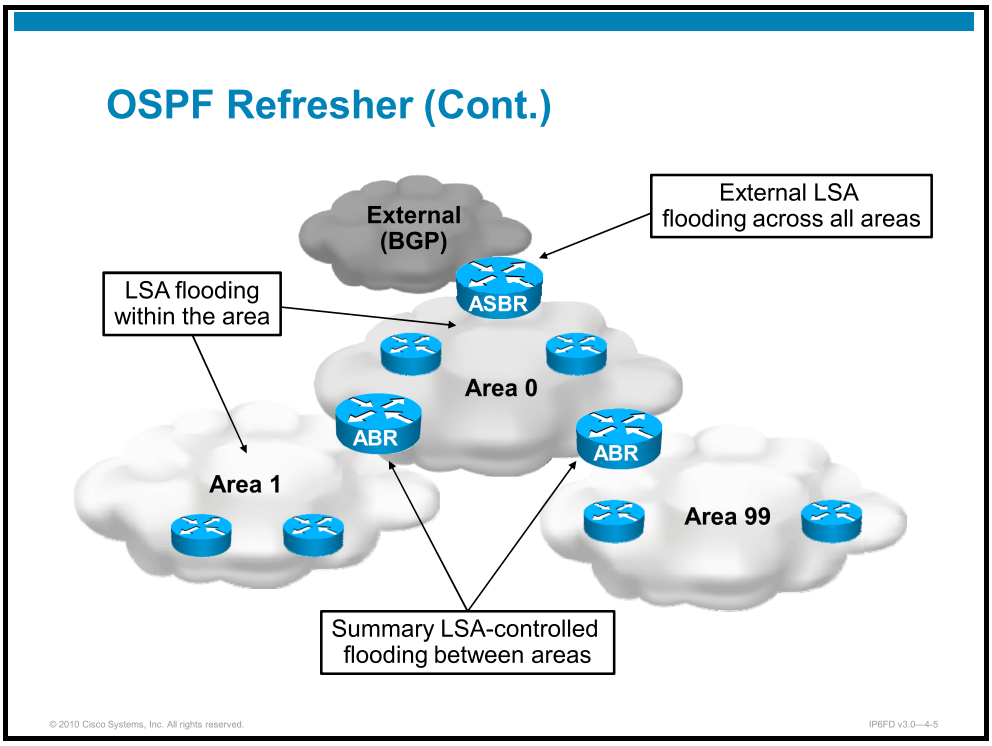
OSPF is a link state protocol. This means that every router has a full topology of the network in memory, and calculates best paths independently from other routers, based on its network knowledge.

All OSPF routers send and collect the link state advertisements (LSAs). These are used to send information about the changes in the network. Routers build the network topology based on the collected LSAs.

ABRs have special handling of LSAs. They limit the change flooding to within the area, and let only the LSAs describing inter-area and external routes to flow between the areas. To do this, ABRs need to have topology databases for all participating areas. Within the area, all routers have the same topology table.

Division of networks into areas is a key feature of OSPF to provide scalability. Routing protocols such as EIGRP consider every network to be “flat”, with a change in the topology sooner or later influencing the decisions in the network, unless the network is manually summarized and thus segmented.

The IS-IS routing protocol takes the multi-area approach even further, making the area divisions even more flexible: in OSPF every area needs to connect to a backbone area (area 0). In contrast, IS-IS does not use backbone area, but uses different levels of routing. Level 1 is used inside one area, to build the topology of all prefixes reachable within a single area. Level 2 is used among different areas, to facilitate inter-area routing. The stretch of Level 2 routers in the IS-IS routing domain must be contiguous. They form a virtual backbone area.



The figure outlines the functioning of OSPF as a link state routing protocol, with the network divided into areas.

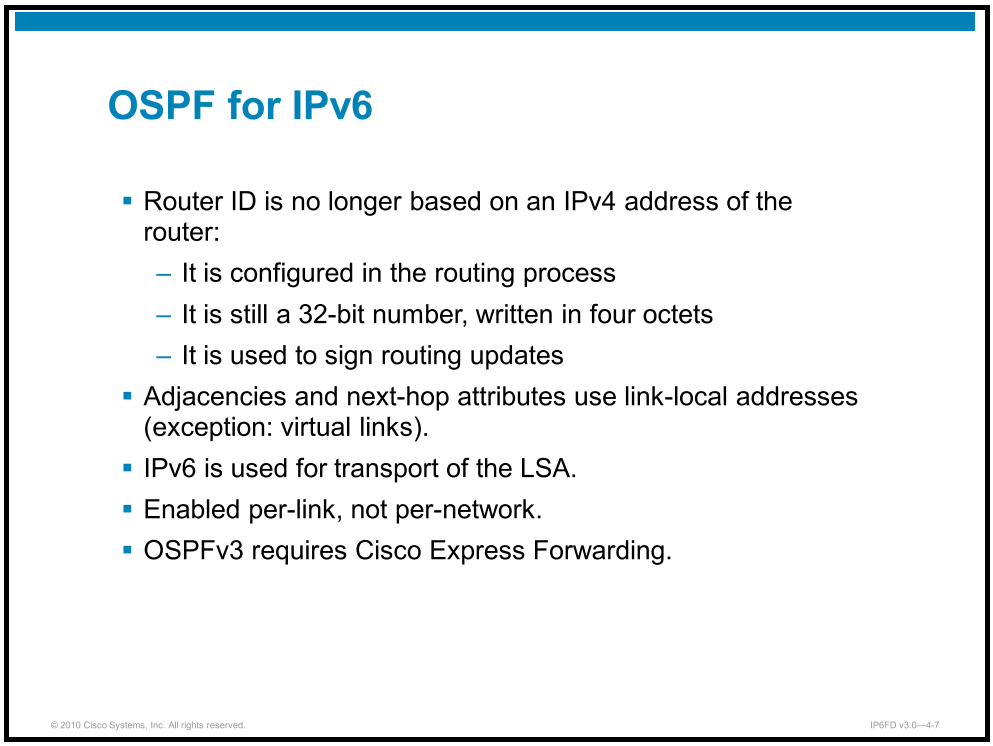
All areas connect to area 0. The interconnecting routers are ABRs. These routers control the flooding of LSAs; some LSAs are allowed to flow within the areas, while other types describe inter-area and external routes.

By using different area types, such as stub and NSSA, you can limit the number of routes in such areas, e.g. allowing only summary, or external, or default routes.

Note For more information about OSPF routing, refer to the Implementing Cisco IP Routing (ROUTE) course.

OSPFv3 Enhancements

This topic describes the changes and enhancements made to OSPFv3 to support IPv6.



These OSPF features have been updated for IPv6:

- In OSPFv2, the router ID is derived from the “highest” IPv4 address of an existing router. It is a general practice to set a loopback interface on the router for the purpose of maintaining the router ID, or setting it administratively in the routing process configuration.
- In OSPFv3, the OSPF process no longer requires an IPv4 address for the router ID, but it does require a 32-bit number to be set.

Note	This 32-bit number is entered as four octets separated by dots [.] and looks like an IP address.

- OSPFv3 no longer carries IP addresses in the OSPF header. The addresses are only carried in the payload.
- OSPFv3 adjacencies use link-local addresses to communicate. Router next-hop attributes are neighboring router link-local addresses (except in the case of a virtual link). Since link-local addresses have the same prefix, OSPF needs to store the information about the outgoing interface.
- OSPFv3 uses IPv6 for transport of LSAs. IPv6 protocol number 89 is used.
- OSPFv3 is enabled per-link and identifies which networks (prefixes) are attached to that link for determining prefix reachability propagation and OSPF area.
- OSPFv3 requires the router to run Cisco Express Forwarding.

OSPF for IPv6 (Cont.)

- Router ID, area ID, and link-state ID remain 32 bits:
  - Not derived from an IPv4 address
- Router LSA and network LSA do not contain IPv4 addresses, these are only 32-bit identifiers.
- LSAs now have a flooding scope defining a radius:
  - Link-local
  - Area
  - Autonomous system
- Handling and forwarding of unknown LSAs is supported—to handle future OSPF extensions.
- Uses IPv6 link-local multicast addresses:
  - FF02::5 OSPF routers
  - FF02::6 OSPF-designated routers

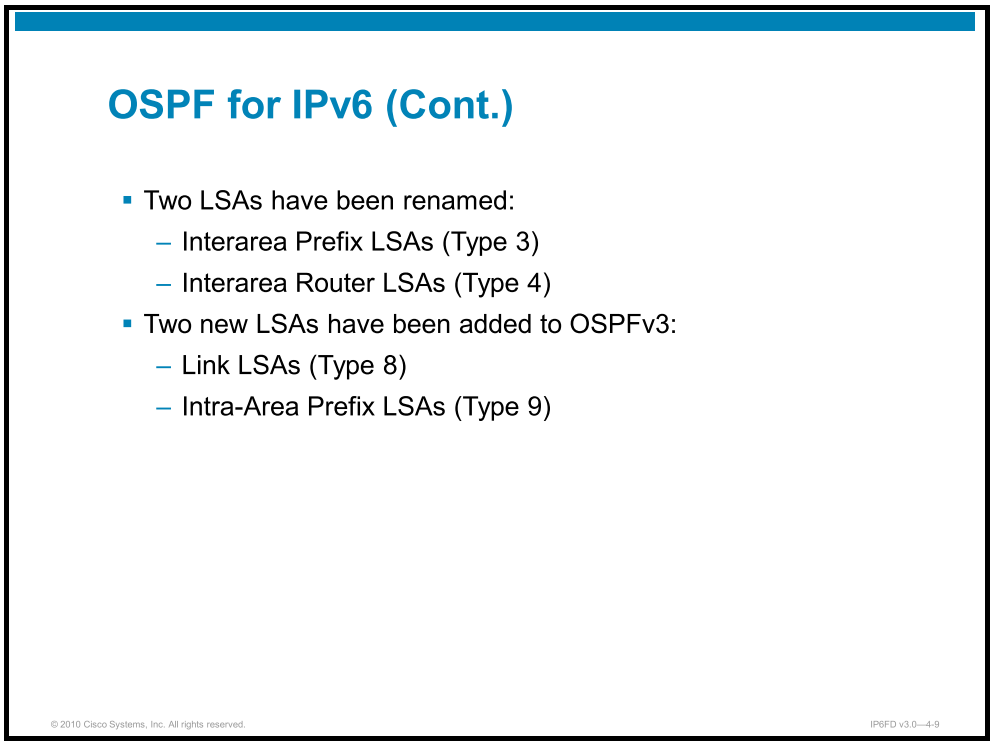
In OSPFv3, the router ID, area ID, and link-state ID are still 32 bits, but these values are not based on IPv4 addresses, though they may look like them. This feature, combined with the removal of addressing from OSPF headers, has made OSPFv3 Layer 3-agnostic. Router LSAs and network LSAs contain only 32-bit identifiers. They do not contain addresses.

LSAs have flooding scopes that define a diameter, as follows:

- **Link-local:** Flood all routers on link
- **Area:** Flood all routers within an OSPF area
- **Autonomous system:** Flood all routers within the entire OSPF autonomous system

OSPFv3 supports the forwarding of unknown LSAs based on the flooding scope. This can be useful in a not-so-stubby area (NSSA). It is also useful when a designated router does not support as many features as other OSPF neighbors, but you still want those OSPF routers to be able to use the new features.

OSPFv3 now takes advantage of IPv6 multicasting, using FF02::5 for all OSPF routers and FF02::6 for the OSPF-designated router (DR) and the OSPF backup designated router (BDR).



There are two renamed LSAs:

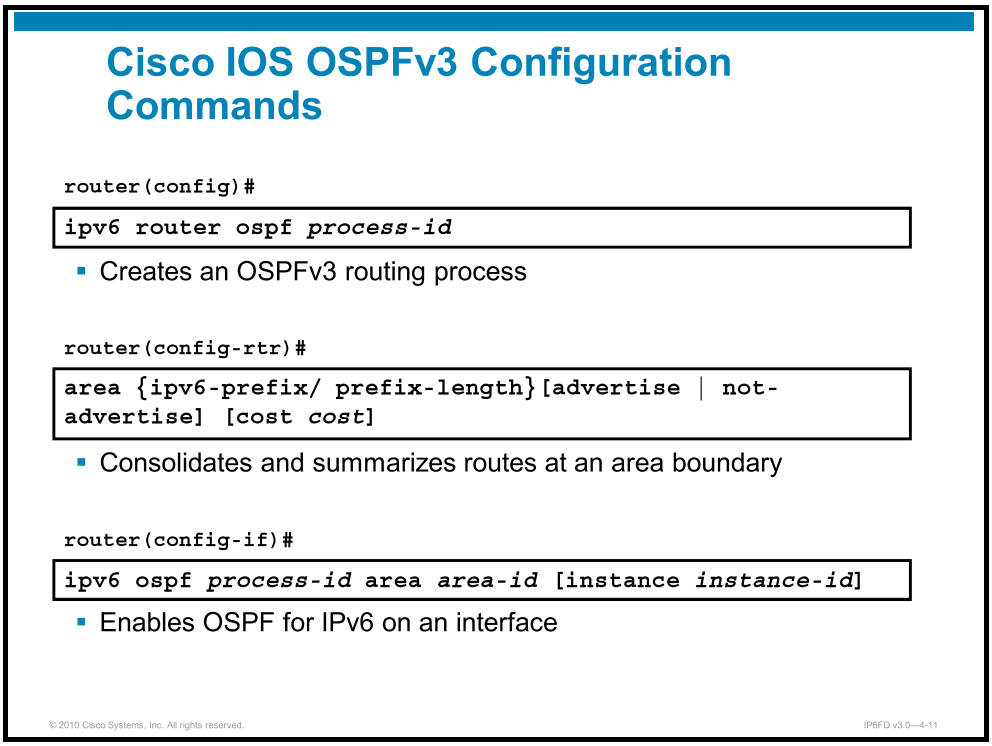
- **Interarea prefix LSAs for Area Border Routers (ABRs) (Type 3):** Type 3 LSAs advertise internal networks to routers in other areas (interarea routes). Type 3 LSAs may represent a single network or a set of networks summarized into one advertisement. Only ABRs generate summary LSAs. In OSPF for IPv6, addresses for these LSAs are expressed as prefix, prefix length instead of address, and mask. The default route is expressed as a prefix with length 0.
- **Interarea router LSAs for Autonomous System Boundary Routers (ASBRs) (Type 4):** Type 4 LSAs 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.

There are two new LSAs in IPv6:

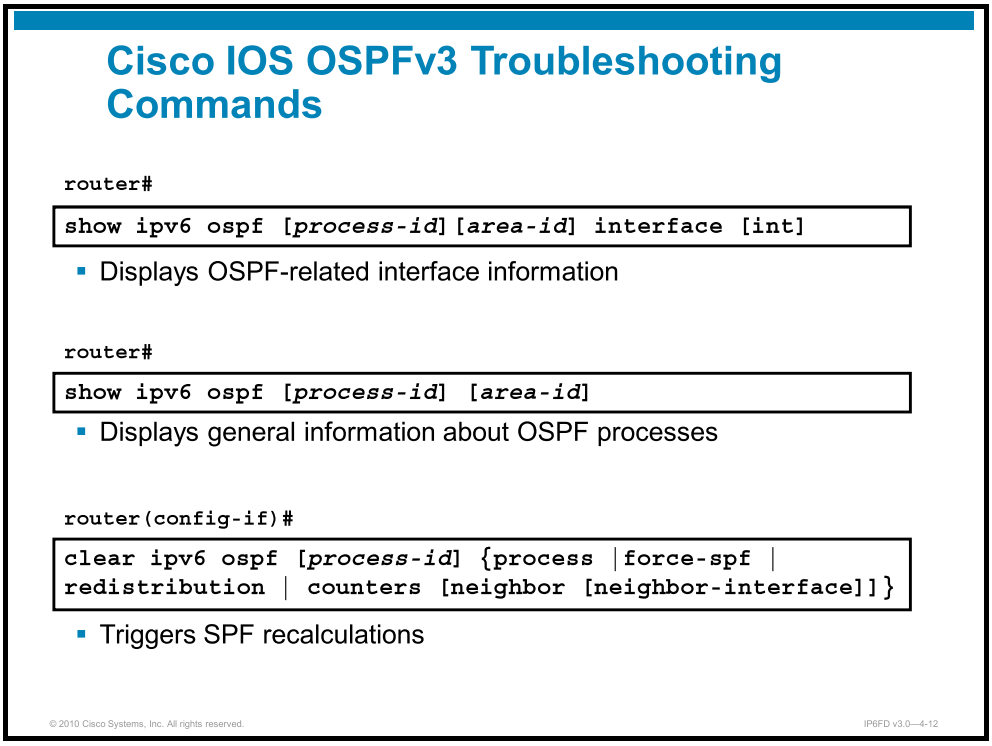
- **Link LSAs (Type 8):** Type 8 LSAs have local-link flooding scope and are never flooded beyond the link with which they are associated. Link LSAs provide the link-local address of the router to all other routers attached to the link. They inform other routers attached to the link of a list of IPv6 prefixes to associate with the link. In addition, they allow the router to assert a collection of option 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. The link-state ID also contains prefixes for stub and transit networks.

OSPFv3 Configuration

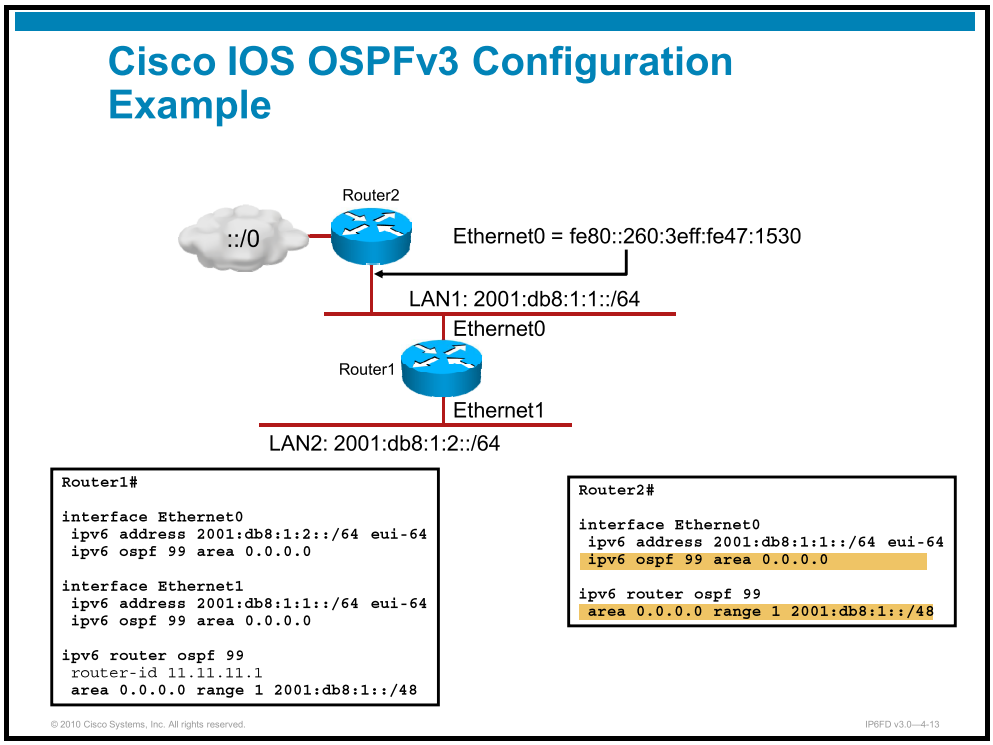
This topic describes how to configure the OSPFv3 protocol on Cisco routers.



The figure shows commonly used Cisco IOS OSPFv3 configuration commands. The syntax is similar, if not identical, to IPv4 counterparts.



The figure shows commonly used Cisco IOS OSPFv3 troubleshooting commands. The syntax is similar, if not identical, to IPv4 counterparts.



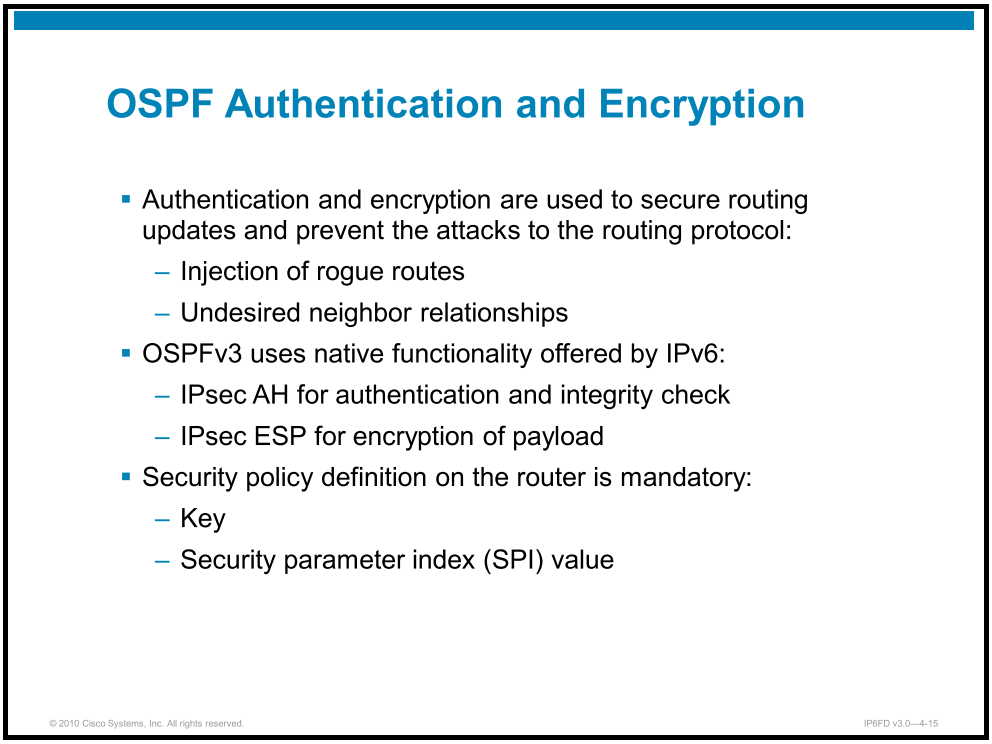
The example shows a single-area OSPF network of two routers. The upper router is connected to the default network. All routers are in Area 0 (area 0.0.0.0). The entire network can be summarized as 2001:db8:1::/48.

The screen text is from the upper router, called Router2. The **ipv6 ospf 99 area 0.0.0.0** interface-specific command creates the IPv6-router OSPF process dynamically.

Tip	The router process description (for example, “ipv6 router ospf 255”) will be created automatically when the interface-specific command is entered. Moreover, additional router-wide parameters are entered under the configuration-level process. These parameters are those related to route redistribution, forwarding on multiple paths (load balancing), route filtering statements, and other (mostly optional) parameters.

OSPFv3 IPsec ESP Authentication and Encryption

This topic describes OSPFv3 IPsec ESP authentication and encryption.



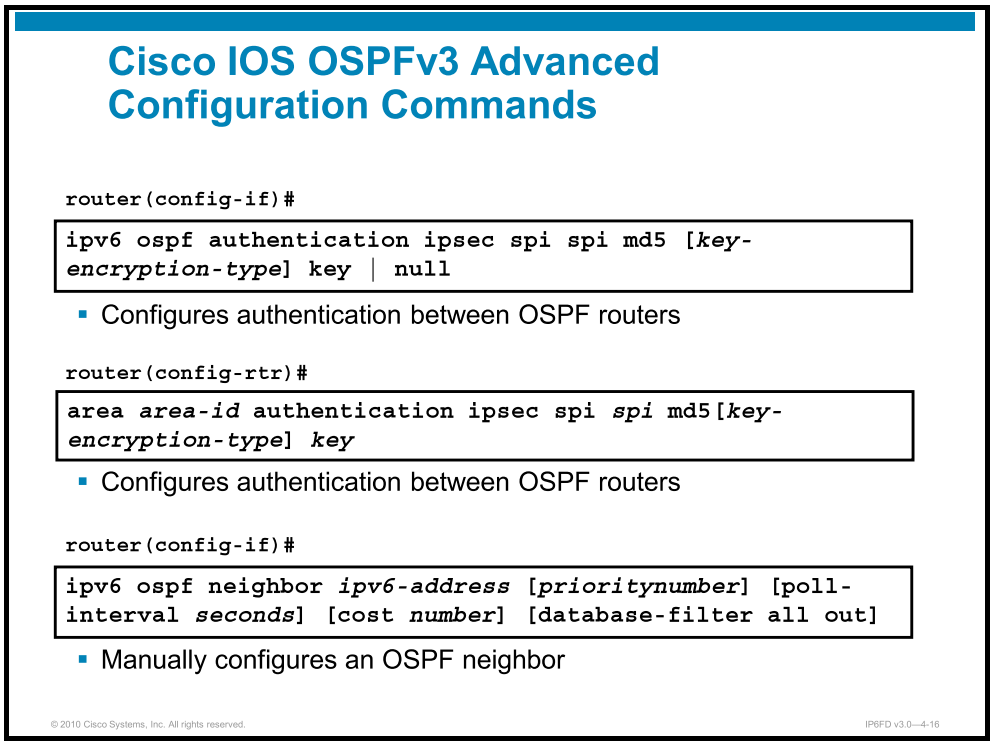
Enhancing routing protocol security is one of the procedures for network and device hardening. OSPFv3 supports authentication and encryption of routing updates.

Using routing update authentication and encryption, you can prevent attacks to the routing protocol. Usually, an attacker might “poison” the routing table of the router by sending a route towards one of the networks using very good cost, and traffic to that network would be diverted to the attacker’s router. Similar applies to attackers injecting the default route.

OSPFv3 uses IPv6 native security capabilities and native security stack. Two possible protocols are available:

- AH for authentication and integrity check
- ESP for encrypting the payload—the routing updates themselves.

Using an IPsec connection requires you to define a security policy for every neighbor router. The security policy defines which protocol is used for communication (AH or ESP), encryption algorithm, the key, and the security parameter index (SPI) value.



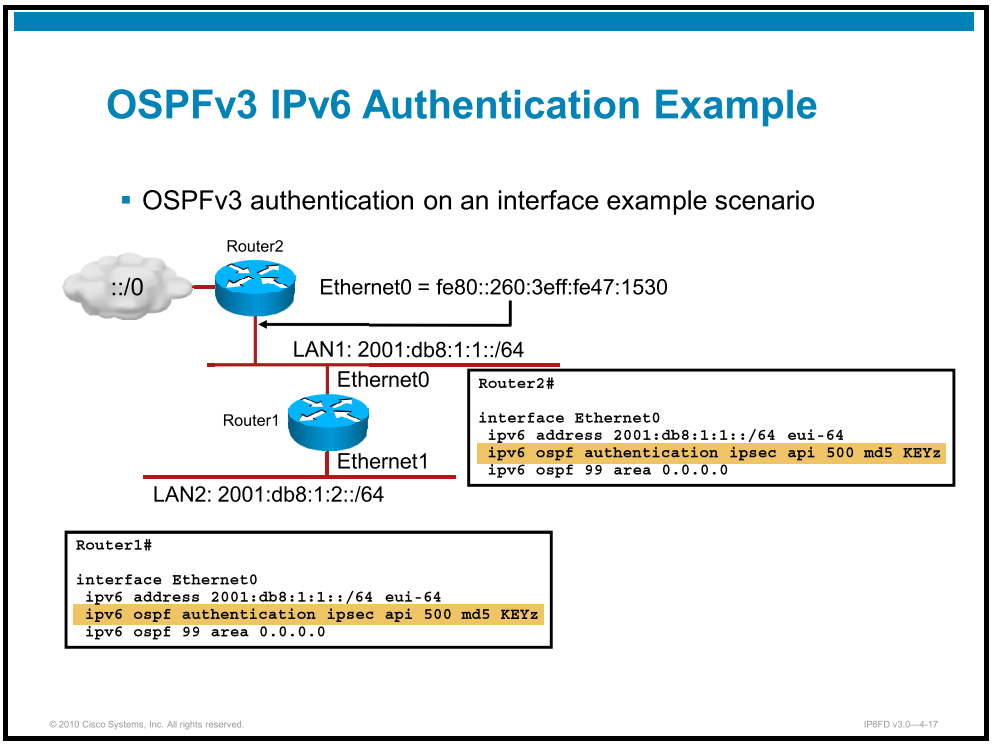
OSPFv3 supports authenticated updates using the underlying IPsec security mechanism of the node. The command specifies that authentication should be enabled and also gives the Security Parameters Index (SPI), authentication algorithm, and shared secret key.

Note	The SPI is used to determine the security parameter index. This is used to identify several IPsec sessions between the same pair of hosts and does not directly apply to OSPFv3; this is required for the security policy to be functional.

To make all routers in a given area authenticate routing updates, you can configure area-wide authentication. This is useful if you have several routers on a broadcast-type link (such as Ethernet), and you do not want to define authentication parameters for every router.

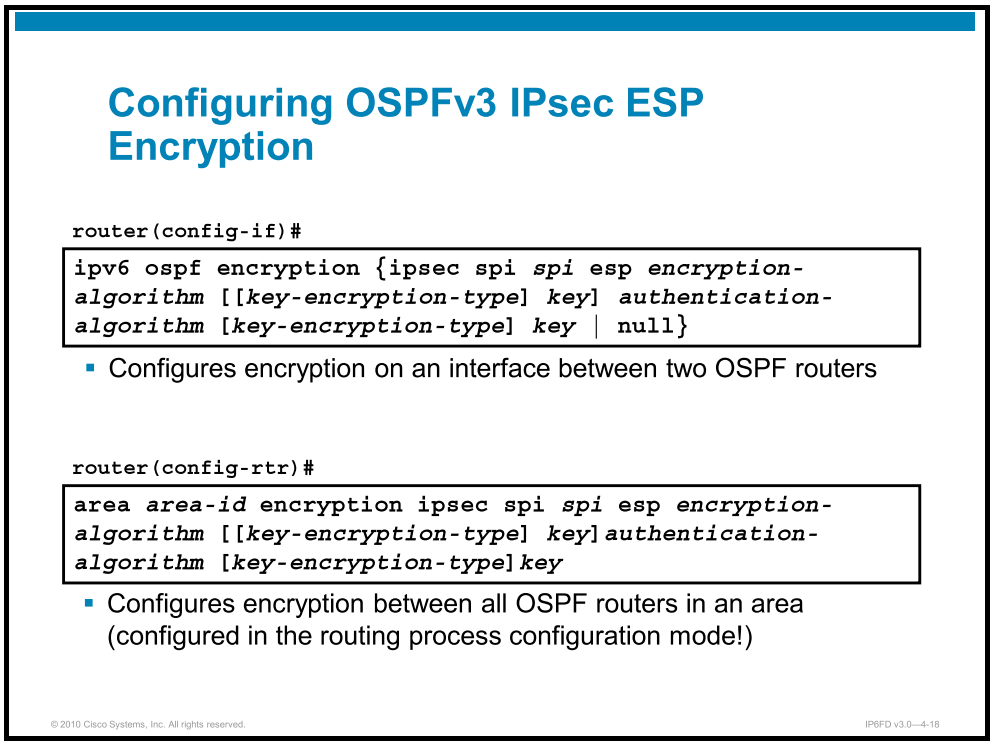
Interface authentication definitions are more useful when you want to authenticate an OSPF session, going either to a service provider (for example, in an MPLS VPN scenario), or going across the WAN (for example, from a hub router to a branch router).

When using nonbroadcast multiaccess (NBMA) in OSPF for IPv6, you cannot automatically detect neighbors. On an NBMA interface, you must configure your neighbors manually using interface configuration mode. The **ipv6 ospf neighbor** command gives the OSPFv3 device the unicast IPv6 address of the peer with which to establish a neighbor relationship.



This example shows the usage and implementation of IPv6 authentication between OSPF routers. The highlighted statements specify the authentication parameters to use for OSPF. These statements can be applied to OSPF areas, in which case the authentication is applied to all interfaces in the area. Or, they can be applied to specific interfaces, as shown here. The Cisco IOS image must support IP Security (IPsec) encryption. OSPF authentication supports both Authentication Header (AH) and Encapsulating Security Payload (ESP).

Tip OSPFv3 has had the OSPF-specific security mechanism used in OSPFv2 (for IPv4) removed and can only be secured using IPsec. This is an important example of how making IPsec support mandatory for IPv6-capable nodes simplifies implementation for other protocols.

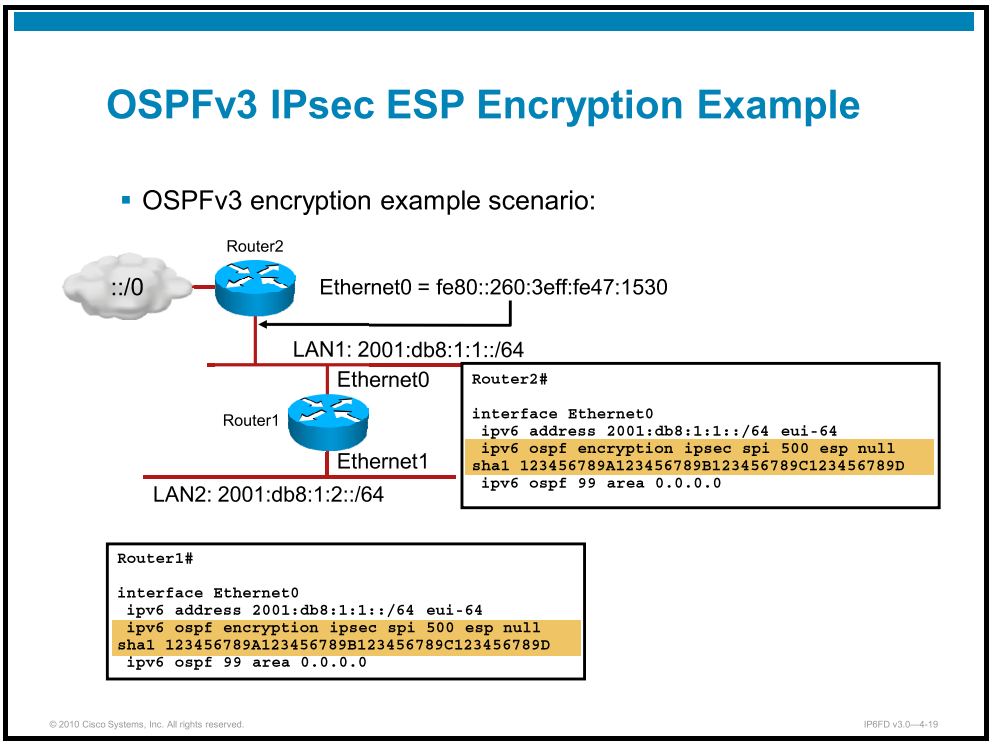


You can configure IPsec encryption for OSPFv3 in a similar way as you configure authentication.

To configure OSPFv3 encryption over one single link, you can use the **ipv6 ospf encryption** interface configuration mode command. The command will enable encryption only for the neighbor router(s) on that interface.

The **area encryption** command enables encryption of routing updates between all routers within an area. This command is useful when there are many neighbors and provides a more elegant way to enforce encryption for routing updates.

Packets that do not pass validation (decryption using the correct key, proper authentication, etc.) are discarded by the router. For a legitimate router neighbor relationship to be established, LSA packets must be validated to be accepted by the router.

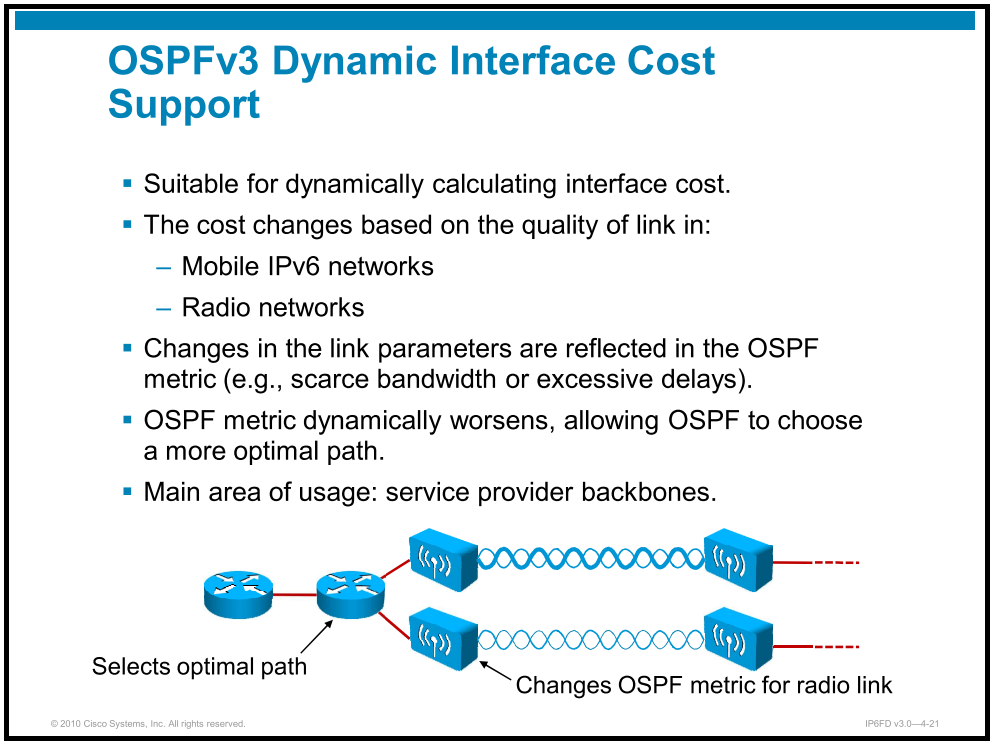


In this example, OSPFv3 encryption is configured on the interfaces between the two routers. IPsec ESP is used as the encrypting algorithm, and SHA-1 is used as a hash algorithm to provide integrity and authentication services.

Note This is inherited from the IPsec protocol suite – packet encryption encapsulations, such as ESP (encapsulated security payload) are used together with packet integrity checking and packet authentication. Suitable algorithms for ESP encryption are AES and variants of DES; suitable algorithms for integrity checking are SHA-1 and MD5.

OSPFv3 Advanced Functionalities

This topic describes OSPFv3 advanced functionalities, such as dynamic interface cost support, fast convergence, and graceful restart.



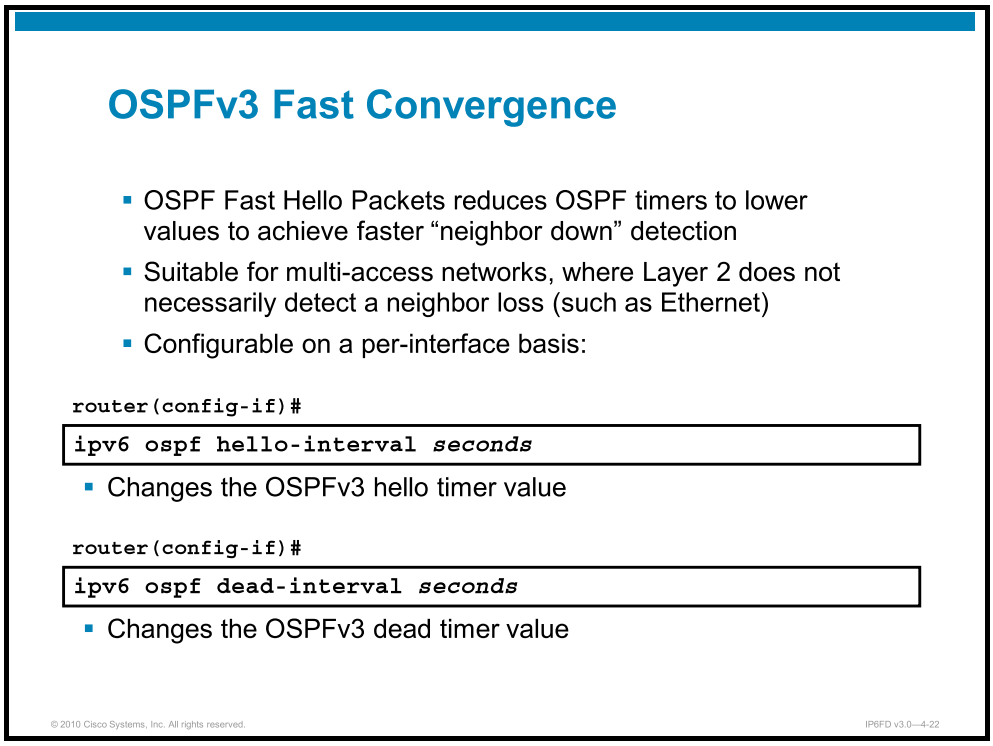
The OSPFv3 dynamic interface cost support is an advanced feature, suitable for cost calculations in mobile IPv6 networks and radio networks. It is of much use for service providers, who utilize OSPF in their backbone network, and who have a large number of IPv6 devices as clients and as gateways. Very large mobile operators and wireless ISPs are the target users for this feature.

The quality of a radio link has a direct impact on the throughput that can be achieved by router-router traffic. These radio links between routers appear as Ethernet links and use PPP over Ethernet (PPPoE) as the underlying protocol. 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 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
- **Current data rate:** The current data rate achieved on the link
- **Latency:** The transmission delay that the packets encounter
- **Resources:** A percentage that can represent the remaining amount of a resource (such as battery power of the radio device)
- **Relative Link Quality:** A numeric value representing relative qualityOSPFv3 then chooses the link which has the best metric to the desired destination, that is, that has the least deductions for link quality issues.

Note For more information, refer to the “OSPFv3 Dynamic Interface Cost Support” listed in the Resources section, and to **Mobile Ad Hoc Networks for Router-to-Radio Communications** (http://www.cisco.com/en/US/docs/ios/ipmobility/configuration/guide/imo_adhoc_rtr2rd_ps6441_TSD_Products_Configuration_Guide_Chapter.html#wp1318087).



The OSPF fast convergence feature is suitable for environments that require very fast reaction upon network failures. Link state protocols are particularly good at this because they maintain a full topology table in the router memory. When a path is lost, the SPF algorithm is rerun to determine another path.

In OSPF, neighbor adjacency is maintained by the periodic transmission of “hello” packets. A neighbor should be declared unavailable if it does not transmit a “hello” packet within a certain time interval. In OSPF, it is usually referred to as the "dead-interval", which usually lasts for three “hello” periods. Therefore, a router neighbor is declared “dead” upon three lost “hello” packets.

OSPF fast convergence is achieved by reducing the OSPF timers for “hello” and “dead” timers. When these are reduced to sub-second values, OSPF can converge in below than one second.

This feature is particularly useful on Ethernet and similar links, when the neighbor down situation does not result in a link down on the interface (two routers connected through a switch—the Ethernet interface will remain “up”, instead of the neighbor router being offline).

Suitable environments for this feature are the data centers, where maximum high availability is required.

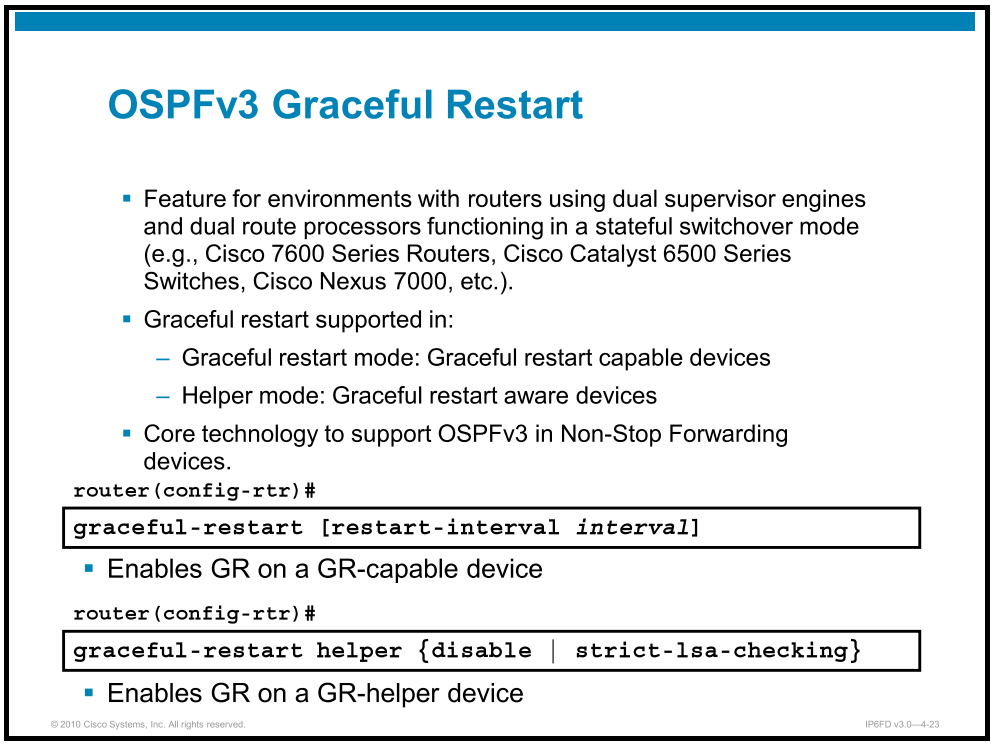
The following commands reduce the OSPF timer values (configured under interface configuration mode):

```
Router(config-if)#ipv6 ospf hello-interval seconds
Router(config-if)#ipv6 ospf dead-interval seconds
```

The Cisco IOS 12.2S special purposes version train allows setting even sub-second timer values.

Note Bear in mind that reducing the timers adds extra load to the CPU of the router, since it must generate and process OSPF hello packets more often.

The network designer should choose what the direction to achieve fast convergence is: either through Non-Stop Forwarding when normal timer values are used and convergence happens on hardware, or using reduced timers and relying on the routing protocol to achieve convergence.



The OSPFv3 graceful restart feature is intended to work on devices with dual supervisor modules or dual route processors. These include highly available modular routers and modular switches, such as the Cisco 7600 Series Routers, Cisco Catalyst 6500 Series Switches, Cisco Nexus 7000 Series Switches, Cisco ASR 1000 Series Routers, etc. On such devices, one of the supervisors operates as active, and the other supervisor operates as “hot-standby”. Routing operations are performed by the active supervisor and this includes maintaining the routing table, and the neighbor relationships with other routers.

In case of any primary supervisor failure, the system switches to the secondary supervisor. The secondary supervisor takes over and relies to the data plane to switch the packets in hardware, until it recreates the routing information base (RIB, that is, the routing table).

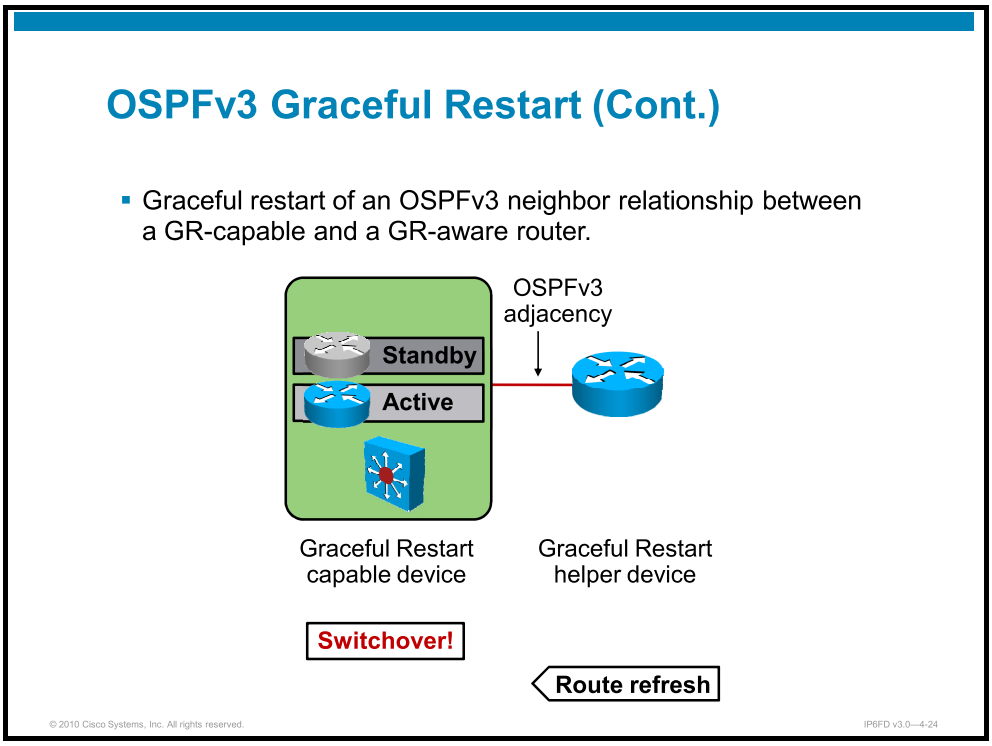
Normally, a switchover to another supervisor module would cause an OSPF neighbor relationship drop, and, consequently, a route flap further in the network.

To prevent the neighbor relationship drop, the OSPF graceful restart keeps the neighbor relationship still established, and requests a “route refresh” from the adjacent router. The recovering router is in **graceful restart mode** and is graceful restart capable, the adjacent neighbor router is in **helper mode** and is a graceful restart aware device.

The OSPF graceful restart feature supports the Non-Stop Forwarding (NSF) capability of highly available routers and switches.

The **graceful-restart** command enables graceful restart on the graceful restart capable router. The **graceful-restart helper strict-lsa-checking** command enables graceful restart on the graceful restart aware router.

Note These two commands run on Cisco IOS Software Version 15.0.



The graphic shows a graceful restart of an OSPF session occurring on a highly available switch.

The redundant supervisor, when becoming the new active, has to build all the software and hardware tables upon taking over the operation. These include the RIB and the routing table, and then all the hardware structures, such as the forwarding information base (FIB).

During the OSPF process recovery and re-learning of the routes from adjacent routers, the router relies on the hardware forwarding tables (FIB) generated by the previous active supervisor.

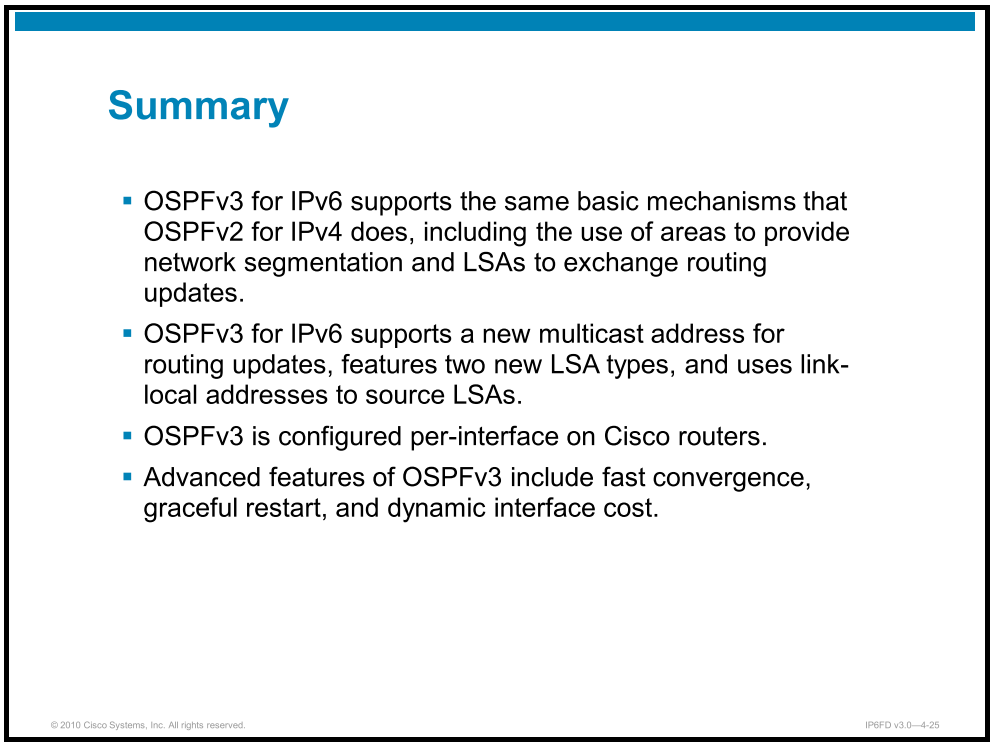
When the router learns new information using the graceful restart, the new routing information is updated in the RIB and in the FIB.

The adjacent router (GR helper), thanks to the graceful restart awareness, does not drop the adjacency, instead in just sends a route refresh.

Note The prerequisite for the graceful restart and NSF is that the router has dual supervisors and hardware-based switching of packets, such as Cisco Express Forwarding (CEF). This is available on the policy feature card (PFC) and the distributed feature cards (DFCs) on the Cisco 7600 Series Routers and the Cisco Catalyst 6500 Series Switches, and on the M1 forwarding engine on the Nexus 7000 Series Switches.

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 3

Examining Integrated IS-IS

Overview

The Intermediate System-to-Intermediate System (IS-IS) is a routing protocol which is being used mainly in service provider environments. Enhancements to IS-IS for IP version 6 (IPv6) enable use of the protocol in transitional networks—a critical requirement for IPv4 and IPv6 interoperability. This lesson describes the IS-IS protocol, including concepts, operations, configuration, and commands.

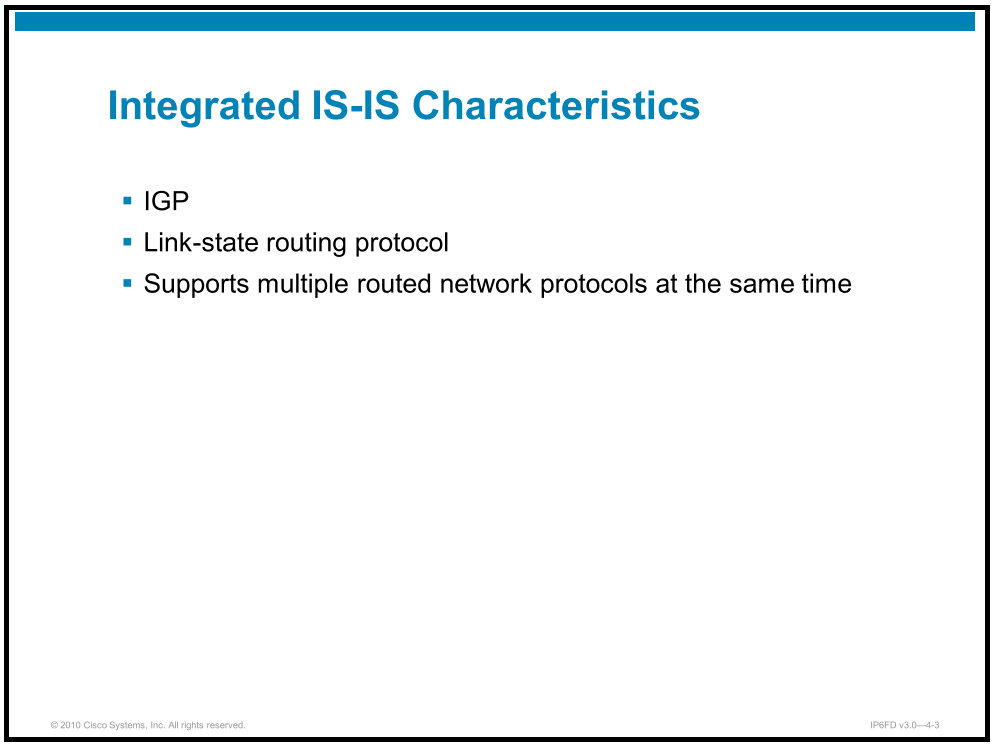
Objectives

Upon completing this lesson, you will be able to describe and configure Integrated IS-IS for IPv6. This ability includes being able to meet these objectives:

- Describe how the IS-IS routing protocol is supported in IPv6
- List the changes made to IS-IS to support IPv6
- Explain the implications of running IS-IS in a single SPF architecture
- Describe multitopology IS-IS for IPv6
- Configure and troubleshoot the IS-IS protocol on Cisco routers

Integrated IS-IS Characteristics

This topic describes how the IS-IS routing protocol is supported for IPv6.



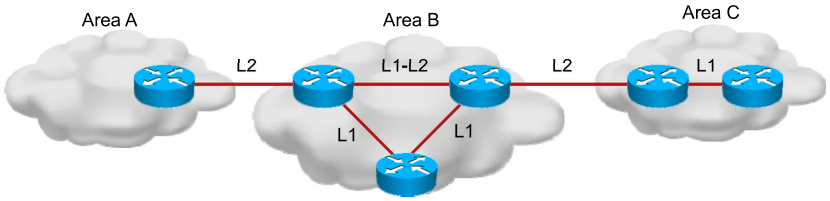
Intermediate System-to-Intermediate System (IS-IS) is part of the interior gateway protocol (IGP) family, which includes Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP). IS-IS is a link-state routing protocol and shares many similarities with OSPF. IS-IS can support link states for multiple routed protocols at the same time, for example, Connectionless Network Service (CLNS), IP, and IP version 6 (IPv6). As discussed later in this lesson, IS-IS can use the same Shortest Path First (SPF) for all network protocols, which raises considerations for network design.

Note IPv6 support in IS-IS is defined in RFC 5308.

Integrated IS-IS Characteristics (Cont.)

Integrated IS-IS Refresher

- Hierarchical routing:
  - No backbone area
  - Level 1 routing within one area
  - Level 2 routing among areas
- Area border on links
- One router belongs to single area
- CLNS used internally



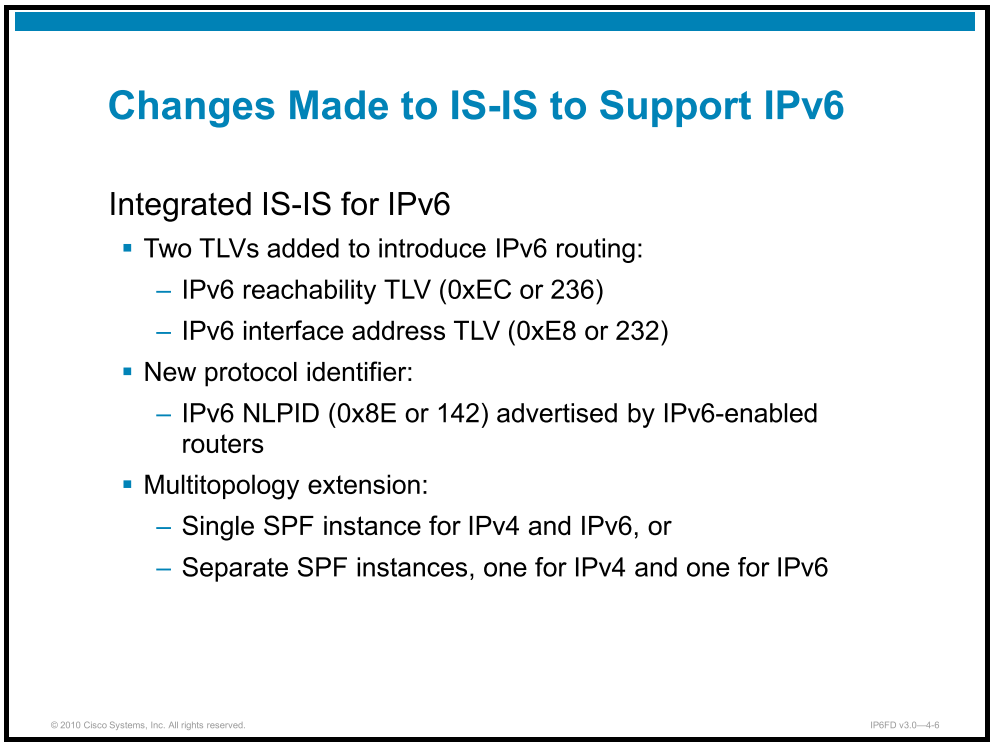
IS-IS, as a link-state protocol similar to OSPF, uses a hierarchical approach to further divide routing domains into areas. In contrast to OSPF, IS-IS does not use a backbone area. It uses different levels of routing. Level 1 is used inside one area to build the topology of all prefixes reachable within a single area. Level 2 is used among different areas to facilitate interarea routing. The Level 2 stretch of routers in an IS-IS routing domain must be contiguous; they form a virtual backbone area. Routers can be Level 1 and Level 2 at the same time to perform both tasks.

IS-IS routers can be part of only one area at a time, area borders lie on links. OSPF has area borders on routers.

IS-IS uses CLNS internally to independently carry routing information from routed protocols.

Changes Made to IS-IS to Support IPv6

This topic describes the changes made to IS-IS to support IPv6.



Two type, length, values (TLVs) are added in IS-IS for IPv6 support. These two TLVs are used to describe IPv6 reachability and IPv6 interface addresses:

- IPv6 reachability TLV (0xEC or 236):
  - Describes network reachability (routing prefix, metric, options)
  - Equivalent to IP internal and external reachability TLVs
- IPv6 interface address TLV (0xE8 or 232):
  - Equivalent to IP interface address TLV
  - For hello protocol data units (PDUs), must contain the link-local address
  - For link-state packets (LSPs), must only contain the nonlink-local address

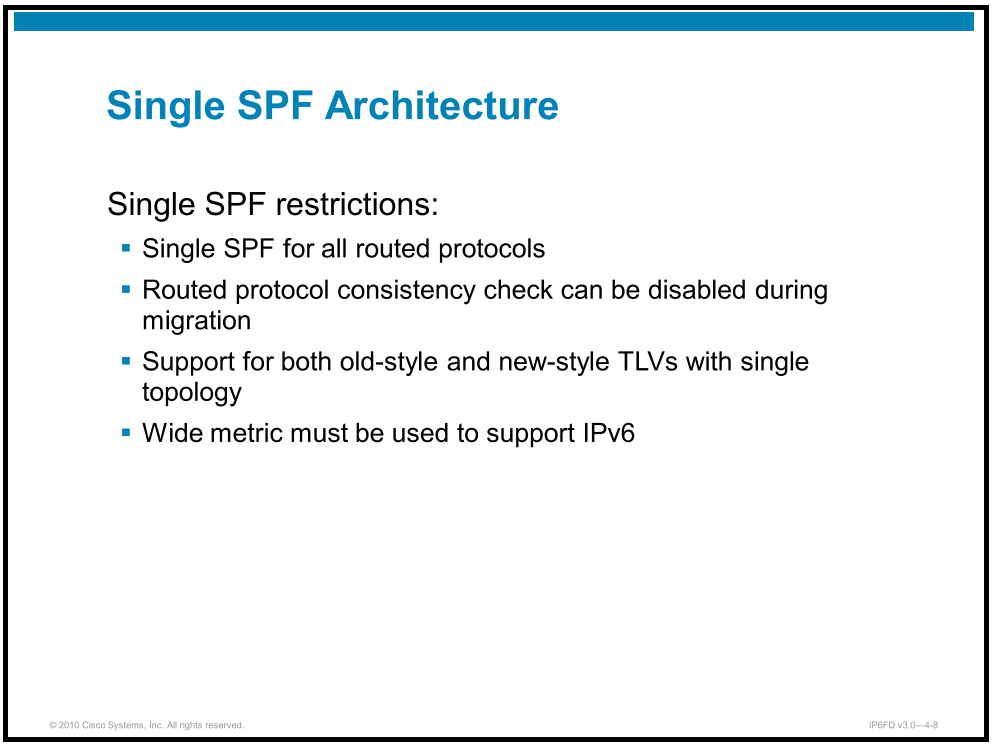
All IPv6-enabled IS-IS routers advertise a Network Layer Protocol ID (NLPID) value of 0x8E (142).

Cisco has added multitopology support to IS-IS to increase flexibility in IS-IS deployment within a dual-stack environment. IS-IS can be deployed using two SPF instances, one for IPv4 and one for IPv6. Multitopology IS-IS provides for some flexibility when transitioning to IPv6. A separate topology is kept for both IPv4 and IPv6 networks, since not all links may be able to carry IPv6 and IS-IS specifically keeps track of those. This way there is a smaller possibility for the traffic to be “black-holed”.

Single topology IS-IS, where there is one SPF instance for both IPv4 and IPv6, also remains a possibility which is even easier to administer, but the network must be homogenous. Same links need to carry IPv4 and IPv6 simultaneously.

Single SPF Architecture

This topic describes the implications of running IS-IS in a single SPF architecture.

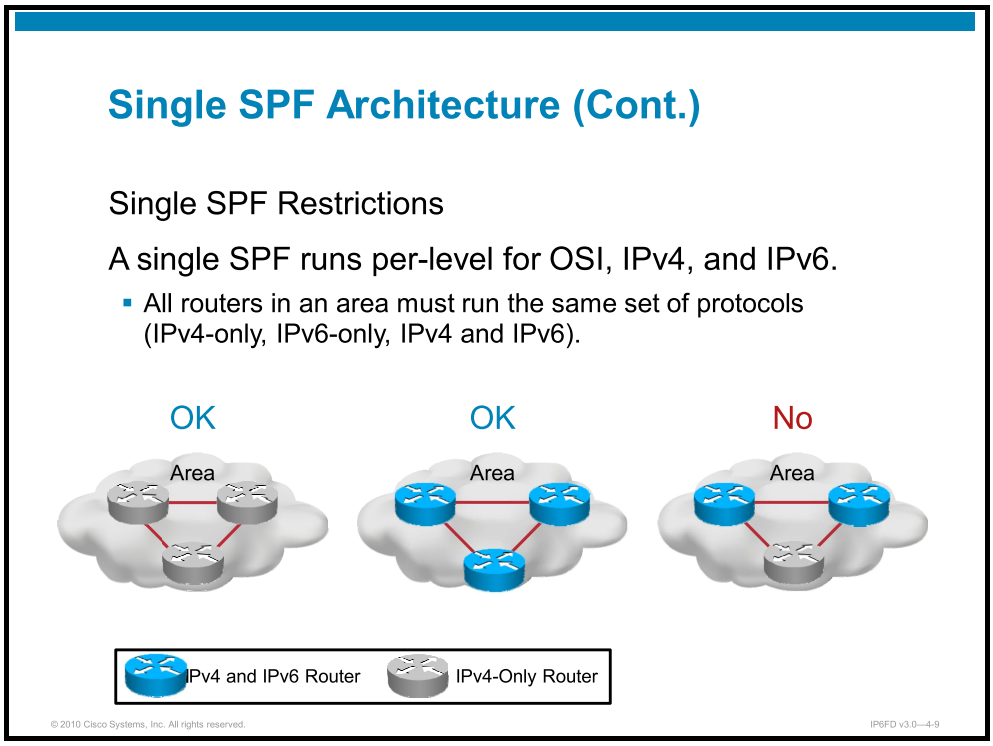


The original design for Integrated IS-IS defines a single SPF for all routed protocols, which adds the assumption that all interfaces included in the routing decisions run all routed protocols.

When migrating from a purely IP version 4 (IPv4) environment to a dual-stack environment, discrepancy in supported protocols would cause adjacencies to fail. To facilitate a seamless upgrade, the engineer may disable adjacency checks during the upgrade to maintain adjacencies active even in a heterogeneous environment.

When single-topology support for IPv6 is employed, either old- or new-style TLVs can be used. However, the TLVs utilized to advertise reachability to IPv6 prefixes use wide metrics, so wide metrics should be used within the whole IS-IS domain.

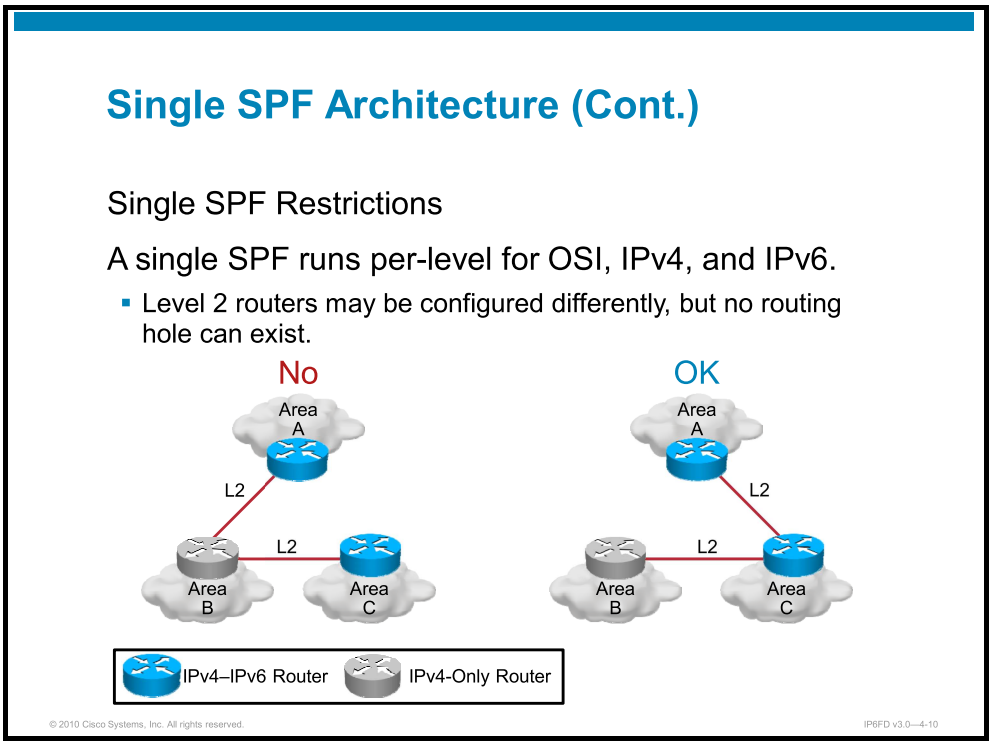
In single-topology IPv6 mode, the configured metric is always the same for both IPv4 and IPv6. The reason for this is that IS-IS establishes routing adjacencies and builds the network topology using CLNS. IPv4 and IPv6 are just routed protocols; for routing information exchange CLNS is used.



The single SPF architecture has the restriction that, within an IS-IS area, all routers must run the same set of protocols.

In some situations, this behavior may be inconvenient, for example, when transitioning an IS-IS IPv4 network to IPv4 and IPv6. Configuring a router from IPv4 IS-IS to IPv4 and IPv6 IS-IS will cause it to drop adjacencies with all its IS-IS IPv4-only neighbors. As discussed later in this lesson, Cisco supports a command that allows a router running IS-IS IPv6 to form an adjacency with a neighbor running IS-IS IPv4 and IPv6.

Tip Suppressing adjacency checking on intra-area links (Layer 1 links) is primarily done during transition from single-topology (IPv4) to multitopology (IPv4 and IPv6) IS-IS networks. Imagine that a large enterprise is integrating IPv6 into their network and it is not practical to shut down the entire enterprise router set for a coordinated upgrade. Without disabling adjacency checking—because routers were enabled for IPv6 and IS-IS for IPv6—adjacencies would drop with IPv4-only routers, and IPv4 routing would be severely impacted. With suppression, IPv6 can be turned up within the enterprise without impacting IPv4 reachability.

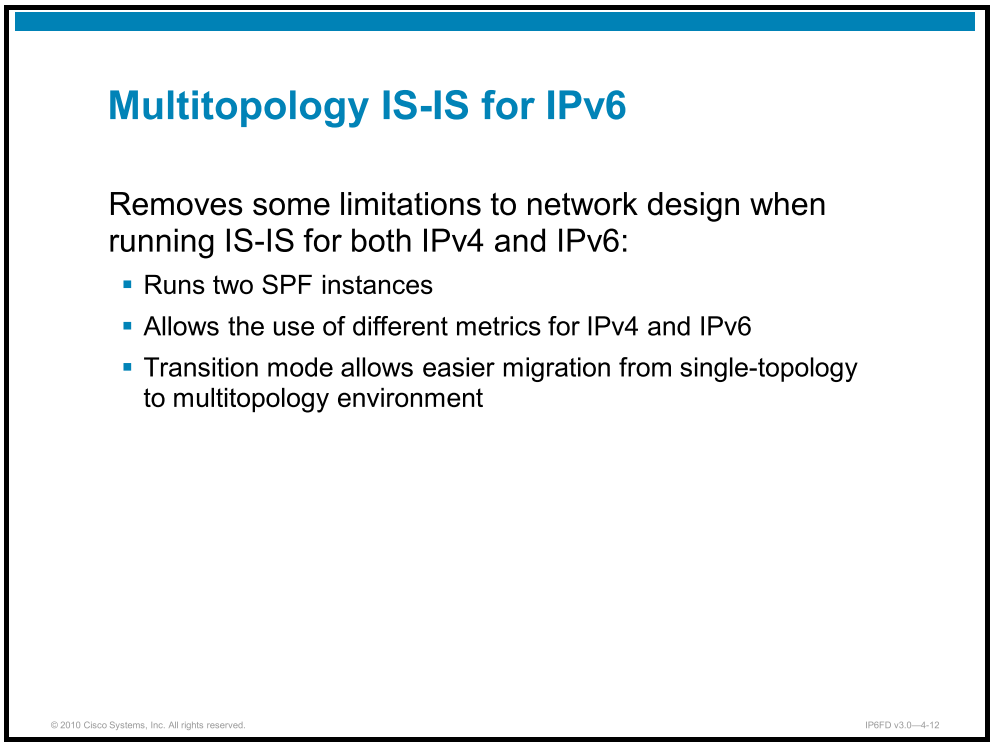


As in any IS-IS network design, Level 2 (backbone) routers must be contiguous. IPv6 adjacency checks are not done on Level 2 links. In the left diagram, the Level 2 routers are not contiguous for IPv6; therefore, this is an incorrect network design. The right diagram shows an example in which the Level 2 routers are contiguous for both IPv4 and IPv6.

Tip This is called a “routing hole.” In the example on the left, adjacencies will be formed across the Layer 2 links between the three areas. However, the IPv6 network is partitioned by the inability of Area B to carry IPv6 traffic. Because IS-IS is managing a single topology, the routers will believe that a path for IPv6 exists across Area B, but all IPv6 traffic sent via that path will fail.

Multitopology IS-IS for IPv6

This topic describes the advantage of running multitopology IS-IS for IPv6.



Multitopology IS-IS for IPv6 is available today and is based on the IETF document *M-ISIS: Multi Topology (MT) Routing in IS-IS* (RFC 5120).

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. Multiple SPFs are performed, one for each configured topology. Therefore, 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 IPv4 and IPv6.

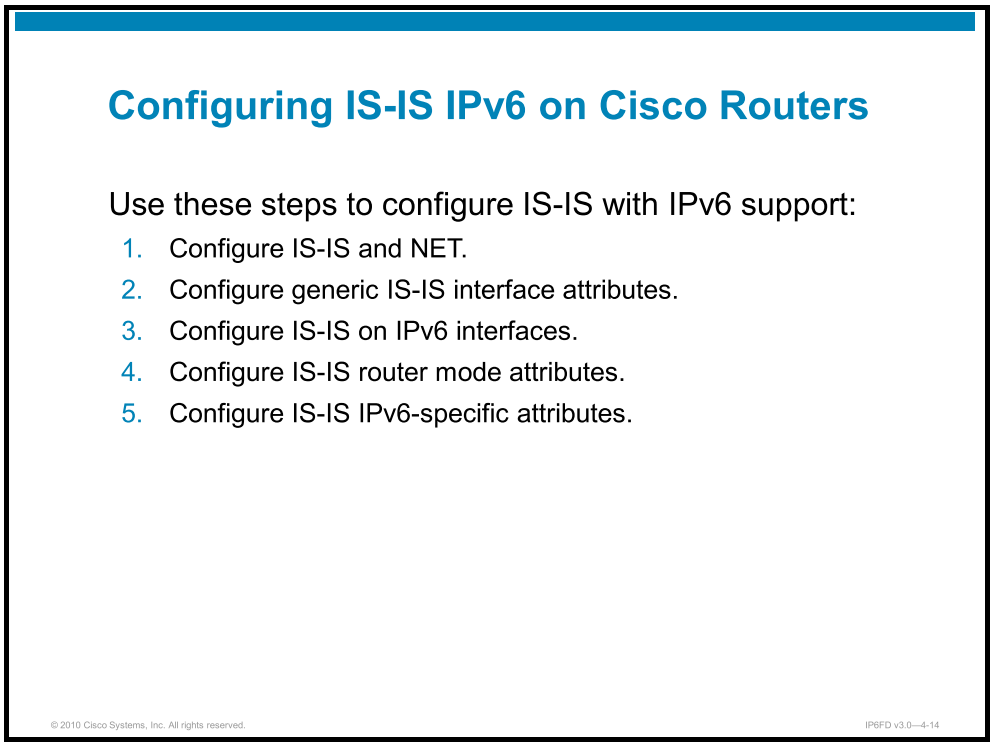
When multitopology support for IPv6 is used, use the **metric-style wide** command to configure IS-IS to use new-style TLVs. TLVs used to advertise IPv6 information in link-state packets (LSPs) are defined to use only wide metrics.

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 routing 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.

IS-IS IPv6 Configuration on Cisco Routers

This topic describes how to configure and troubleshoot the IS-IS protocol on Cisco routers.



Follow these steps to configure IS-IS IPv6 support:

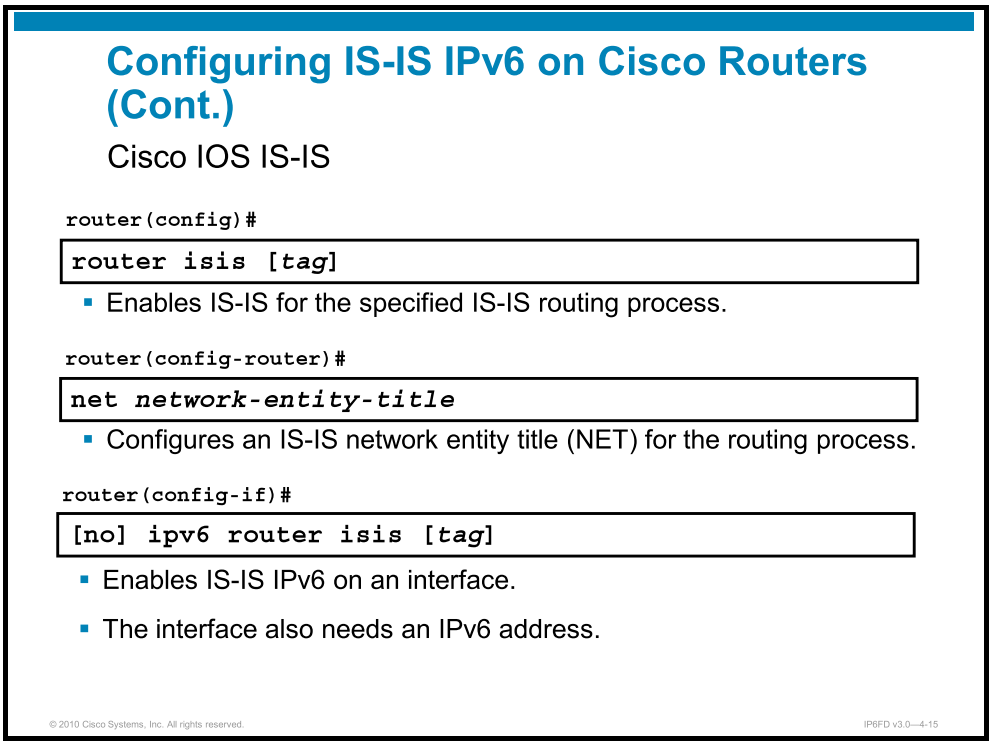
Step 1 Configure the IS-IS routing process and specify the network entity title.

Step 2 Configure generic IS-IS interface attributes using existing IS-IS commands (circuit-type, priority, and so on).

Step 3 Configure IS-IS on interfaces. The interfaces must have the IPv6 protocol stack enabled, for example, having an IPv6 address assigned, or autoconfigured.

Step 4 Configure IS-IS router mode attributes. The majority of IS-IS router mode commands are generic and apply to both IPv4 and IPv6.

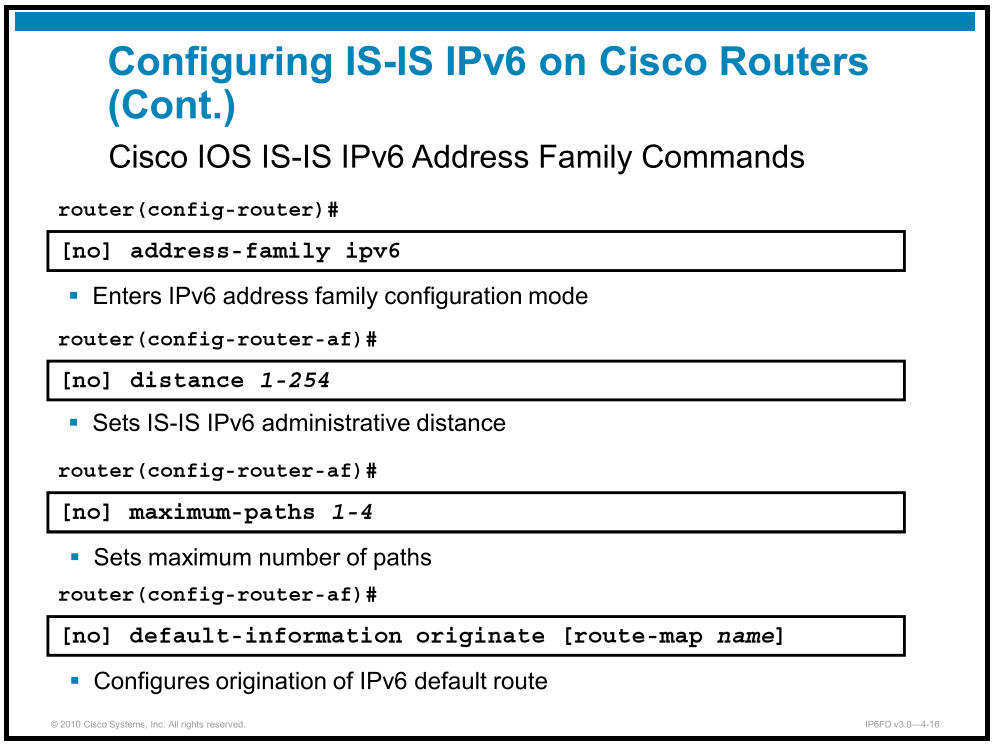
Step 5 Configure IS-IS IPv6-specific attributes. IPv6 attributes are configured via the IPv6 address family submode of the router mode.



Use the **router isis** command to enable IS-IS for the specified IS-IS routing process, and enter the router configuration mode.

The **net** command configures an IS-IS network entity title (NET) for the routing process.

Use the **ipv6 router isis** command to enable IS-IS IPv6 on an interface. Note that the interface must be IPv6-enabled; that is, it must either have an IPv6 address configured or have been explicitly enabled via the **ipv6 enable** command.



Use the **address-family ipv6** command to enter the IPv6 address-family submode from the router mode. IPv6-specific IS-IS attributes are configured under this submode. The **no** form of the command will reset all IPv6-specific configurations to default values.

These IS-IS IPv6 commands and attributes are used under the address-family IPv6 submode and are applied to the IPv6 routing table only:

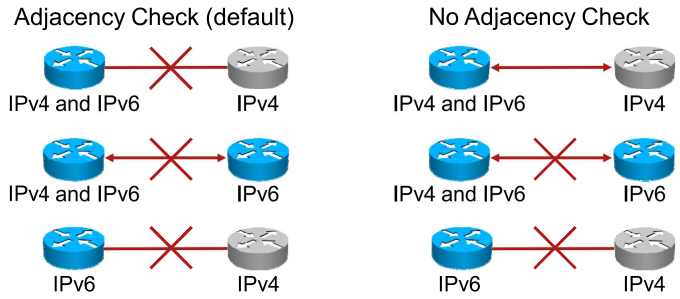
- **distance** *1–254*: Sets the administrative distance of IS-IS IPv6. The default administrative distance for IS-IS is 115.
- **maximum-paths** *1–4*: Sets the maximum number of paths allowed for a route learned via IS-IS IPv6. The default number is four.
- **default-information originate** [**route-map** *name*]: Configures origination of the IPv6 default route (::/0) by IS-IS. It is used in the same manner as the existing IPv4 **default-information** command.

Configuring IS-IS IPv6 on Cisco Routers (Cont.)

Cisco IOS IS-IS IPv6 Adjacency Check

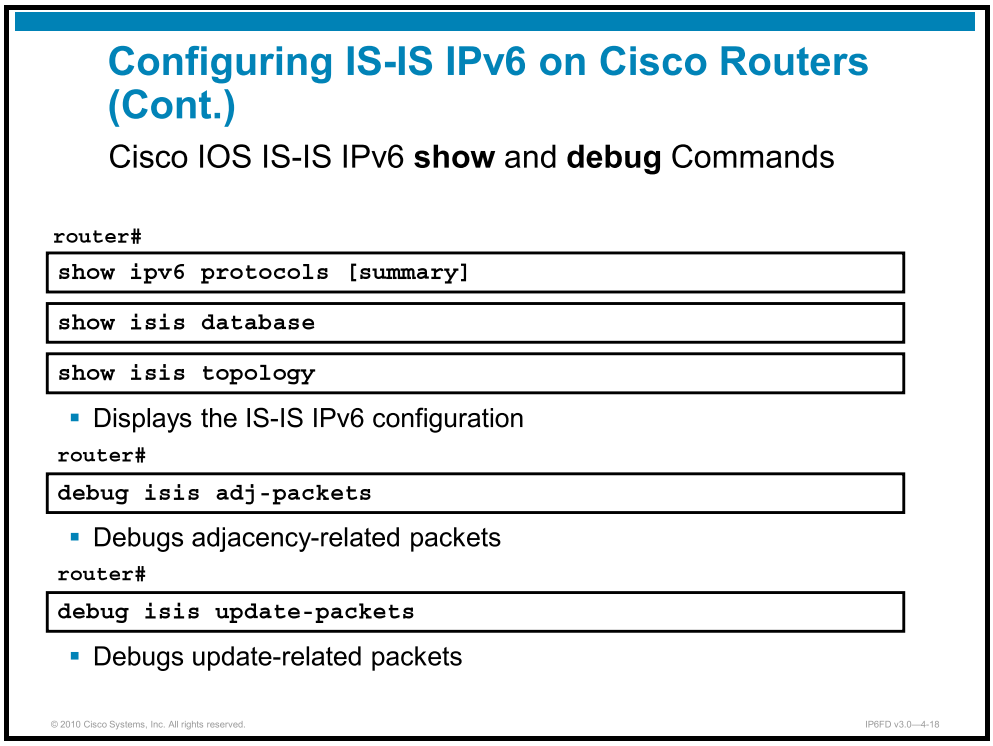
```
router(config-router-af)#
[no] adjacency-check
```

- Enables or disables IPv6 adjacency checks



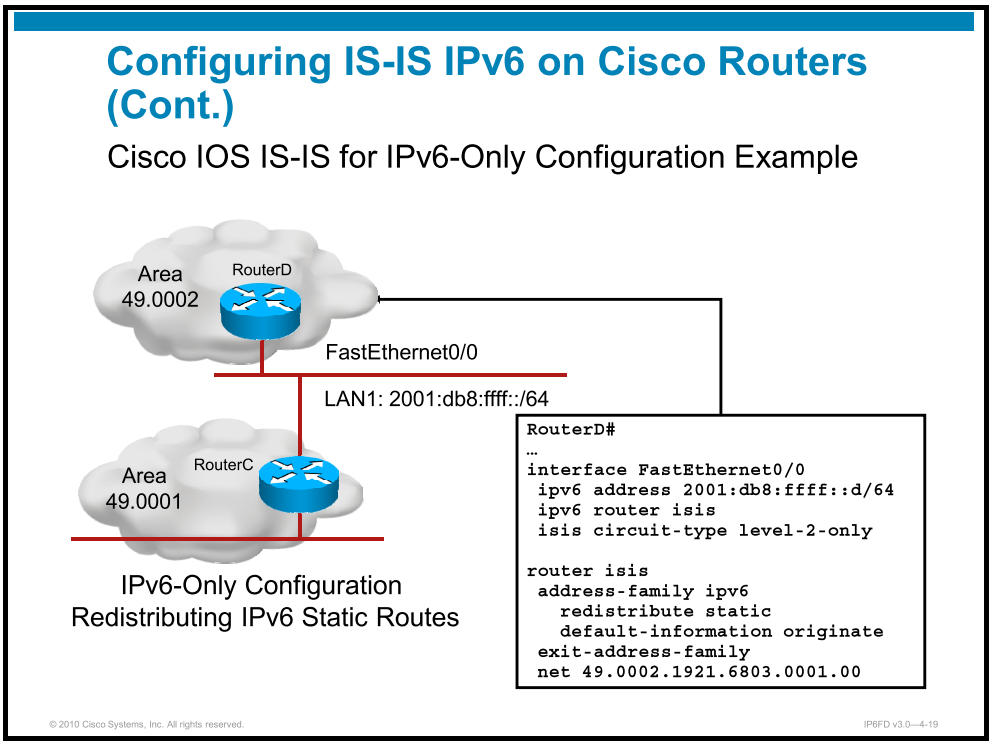
The **adjacency-check** command enables or disables adjacency IPv6 protocol-support checks. If enabled (default), the router will not form an adjacency with a neighbor not supporting IS-IS IPv6.

When enabled, the **adjacency-check** will allow an adjacency to be formed between an IPv4–IPv6 router and an IPv4 router. This configuration may be convenient when transitioning an IS-IS IPv4 network to IPv4 and IPv6. Note that the **adjacency-check** suppresses IPv6 checks only. IS-IS IPv4 also checks the protocol support of neighbors and will not allow an adjacency between a router running IS-IS IPv4 and a neighbor not supporting IPv4. Also, if the IS-IS router determines that the shortest path to an IPv6 destination lies via a non-IPv6 neighbor, the route to the destination will not be installed in the IPv6 Routing Information Base (RIB).



The following partial list of **show** and **debug** commands are not specific to IPv6, but are useful in the context of examining and debugging an IS-IS IPv6 configuration:

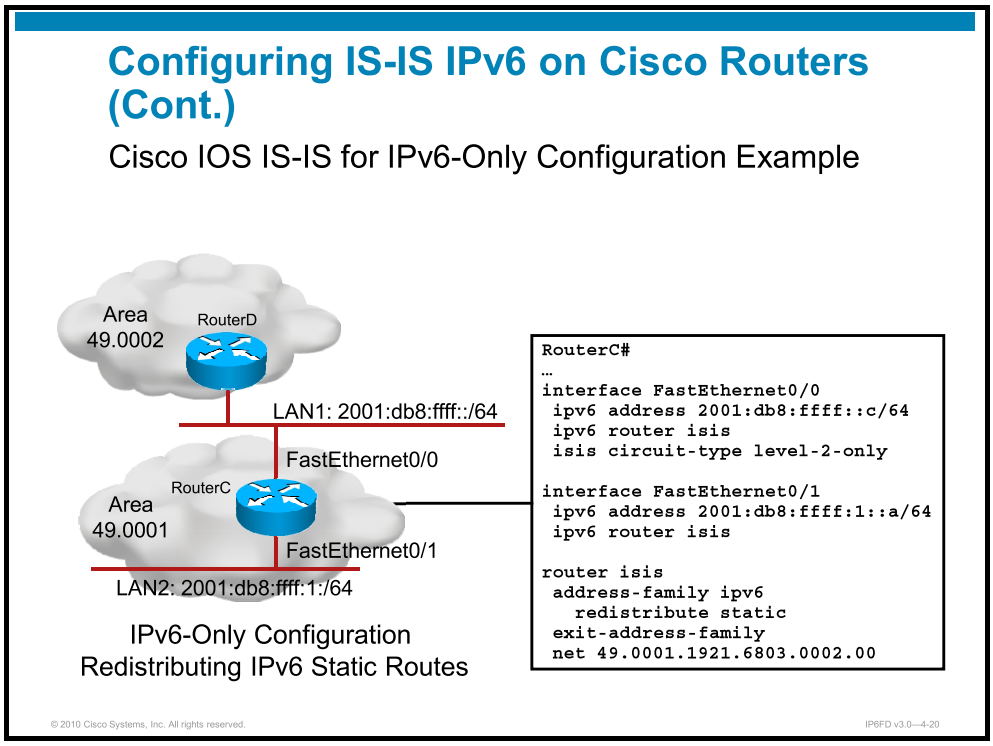
- The **show ipv6 protocols [summary]** command shows the current IPv6 routing protocol configuration.
- The **show isis database** command displays the IS-IS link-state database.
- The **show isis topology** command shows a list of all connected routers in all areas.
- The **debug isis adj-packets** command displays the adjacency packet events.
- The **debug isis update-packets** command displays the IS-IS update packet events.



The example is a two-area network in which the routers are configured as IPv6-only IS-IS routers. Note that only IS-IS configuration is shown (in other words, this is not a complete IPv6 router configuration).

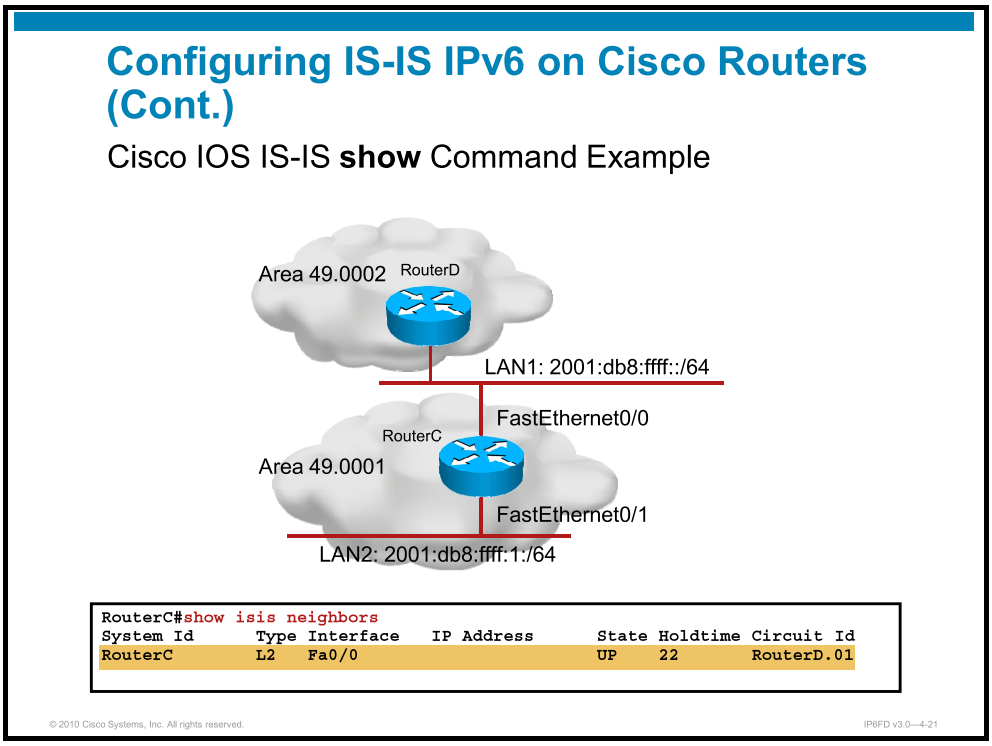
The configuration shows that RouterD has an IPv6 address configured on interface Fast Ethernet0/0, with IS-IS IPv6 enabled using the interface command **ipv6 router isis**. The command **isis circuit-type level-2-only** on Fast Ethernet0/0 configures that interface as a Level 2-only IS-IS interface—a backbone connection.

Under IS-IS router mode (**router isis**), a default route (::/0) is advertised in IS-IS IPv6 using the command **default-information originate** under the **address-family ipv6** submode. The network service access point (NSAP) address on RouterD is specified as net *49.0002*.1921.6803.0001.00 (Area *49.0002*).

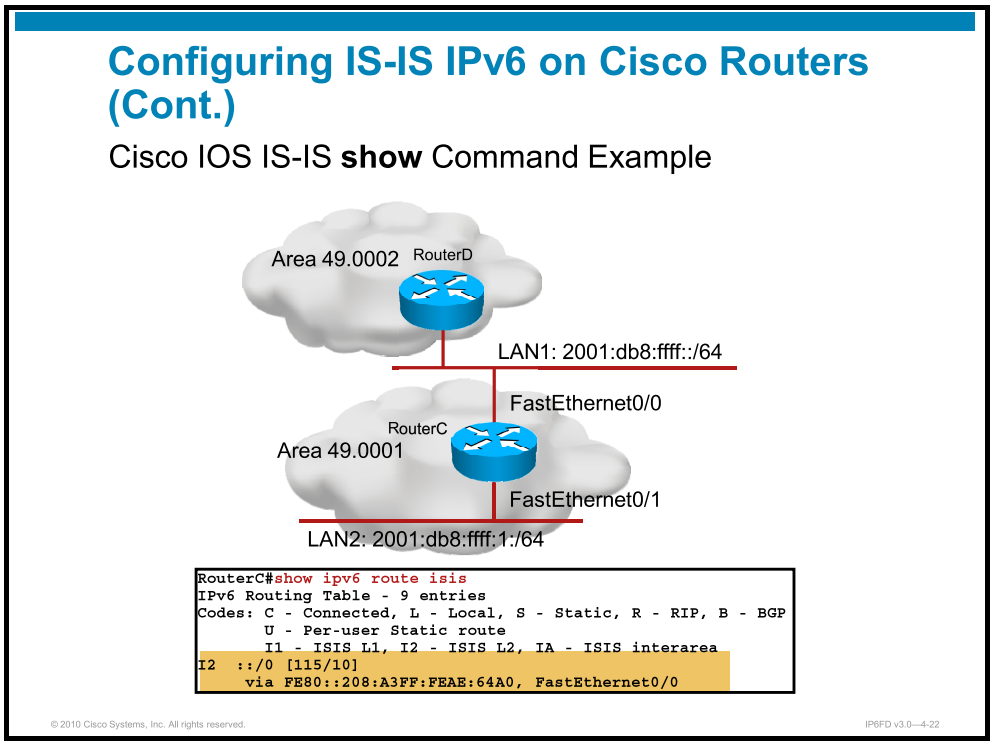


The configuration shows that RouterC has an IPv6 address configured on both Fast Ethernet interfaces with IS-IS IPv6 enabled using the interface command **ipv6 router isis**. The command **isis circuit-type level-2-only** on Fast Ethernet0/0 configures that interface as a Level 2-only IS-IS interface.

Under IS-IS router mode (**router isis**), the static IPv6 routes on RouterC are redistributed into IS-IS IPv6 using the command **redistribute static** under the **address-family ipv6** submode. The NSAP address on RouterC is specified as net *49.0001*.1921.6803.0002.00 (Area *49.0001*).



If IS-IS adjacencies are successfully established during verification, issue the **show isis neighbors** command first. The figure shows the output of this command. In this scenario, because both routers (RouterD and RouterC) are configured as described in previous figures—with Level 2 [inter-area] routing only—only a Level 2 adjacency is formed.



In this example, both RouterC and RouterD are configured as IS-IS IPv6 routers, and RouterD originates the default IPv6 route. This example displays the IPv6 routing table on RouterC, using the **show ipv6 route is-is** command. The default IPv6 route was learned through a Level 2 IS-IS IPv6 update. The link-local IPv6 address of RouterD on the interface facing RouterC is FE80::208:A3FF:FEAE:64A0.

Note In IPv6, routing protocols use both the next-hop link-local IPv6 address and the outgoing interface (as a pair) to define the next router towards a destination network.

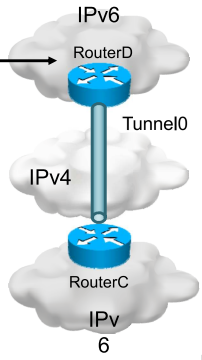
Configuring IS-IS IPv6 on Cisco Routers (Cont.)

Cisco IOS IS-IS for IPv6 over IPv4 Tunnel Configuration Example

IPv6 over GRE IPv4 tunnel configuration:

- GRE tunnel type is required for IS-IS

```
RouterD#
…
interface Tunnel0
 ipv6 address 2001:DB8:ffff::d/64
 tunnel source Ethernet0
 tunnel destination 192.168.99.1
 ipv6 router isis
 tunnel mode gre ipv6

router isis
 net 49.0002.1921.6803.0001.00
 address-family ipv6
 …
```



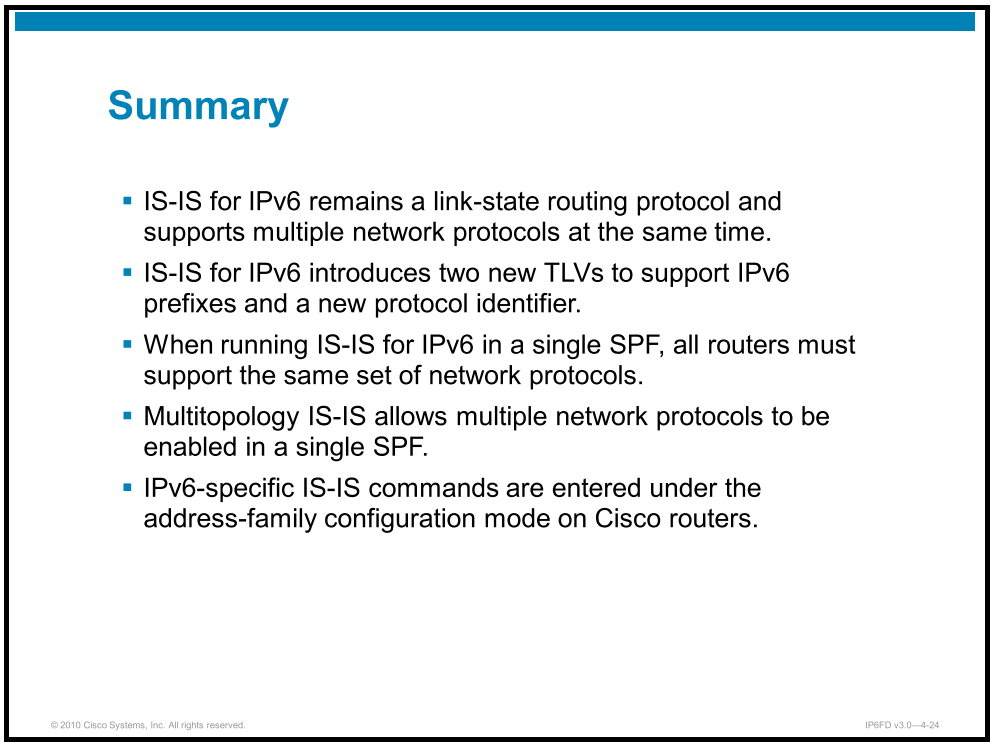
This example shows two distant IPv6 networks in which the routers are configured with a Generic Routing Encapsulation (GRE) tunnel to carry IPv6 inside a GRE IPv4 tunnel. IS-IS cannot be used over an IP-in-IP tunnel because IS-IS is itself a Layer 3 protocol, like CLNS, IPv4, and IPv6.

The configuration shows that RouterD has an IPv6 address configured on the Tunnel0 interface, with IS-IS IPv6 enabled using the **ipv6 router isis interface** command.

Note IS-IS is not supported on manual IPv6-in-IPv4 tunnels, ISATAP tunnels, etc. since these tunnel interfaces are not capable to carry the CLNS protocol, which IS-IS uses for adjacency establishment and routing information exchange. GRE tunnels are able to transport CLNS over IPv4 or IPv6.

Summary

This topic summarizes the key points discussed in this lesson.



References

More information about IS-IS can be found in the following documents:

- *RFC 5308: Routing IPv6 with IS-IS*
  http://tools.ietf.org/html/rfc5308
- *RFC 5120: M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs)*
  http://tools.ietf.org/html/rfc5120
- *Cisco IOS IPv6 Configuration Guide, Release 12.4: Implementing IS-IS for IPv6*
  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-is-is.html

Lesson 4

Examining EIGRP for IPv6

Overview

Although proprietary to Cisco, Enhanced Interior Gateway Routing Protocol (EIGRP) is widely used. Supporting IP version 6 (IPv6) is important for the continued success of EIGRP. This lesson describes Cisco EIGRP, including its operation, configuration, and commands.

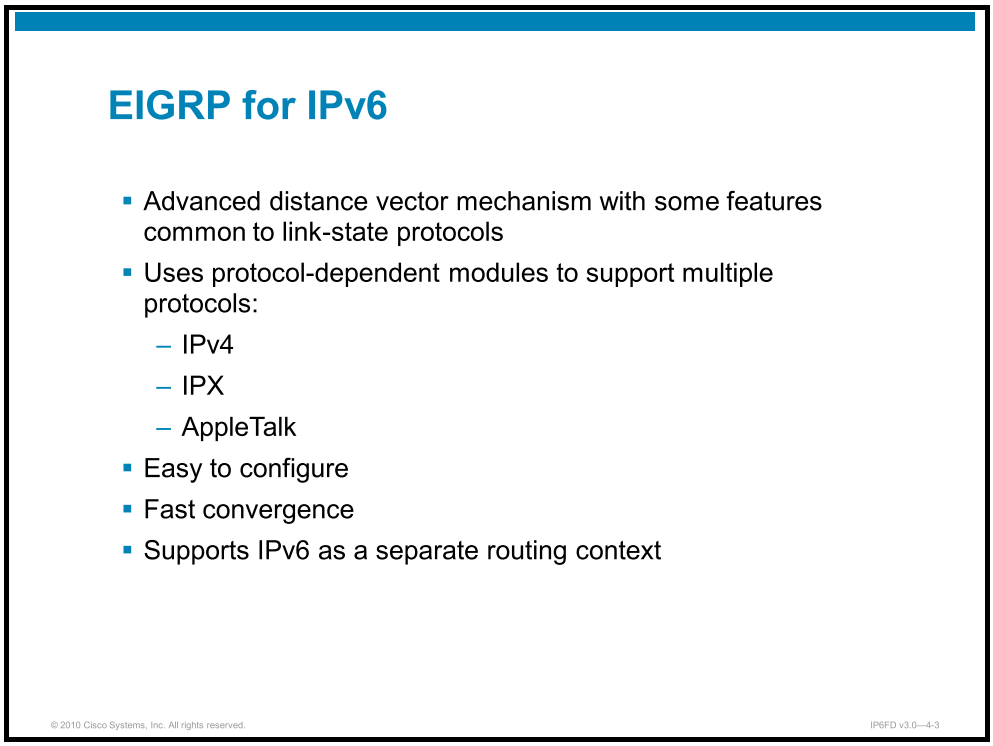
Objectives

Upon completing this lesson, you will be able to describe EIGRP support for IPv6. This ability includes being able to meet these objectives:

- Describe Cisco support for IPv6 routing with EIGRP
- Configure EIGRP for IPv6 on Cisco routers

EIGRP for IPv6

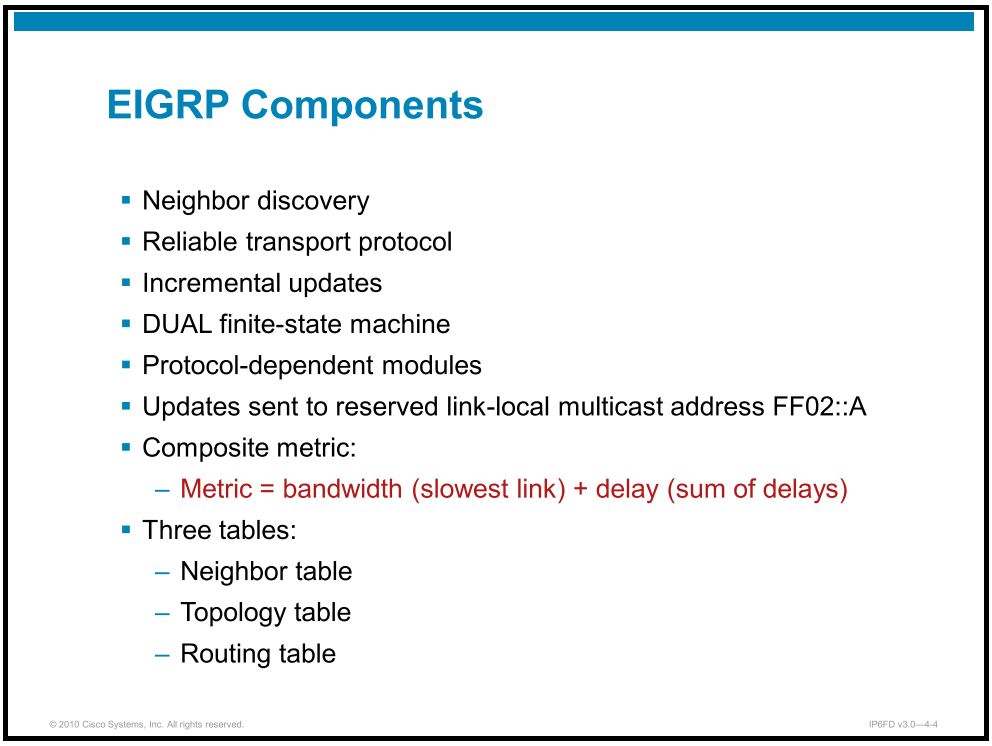
This topic describes Cisco support for IPv6 routing with EIGRP.



Although the configuration and management of EIGRP for IP version 4 (IPv4) and EIGRP for IPv6 are similar, they are configured and managed separately.

EIGRP is inherently a multiprotocol routing protocol, because it has supported non-IP Internetwork Packet Exchange (IPX) and AppleTalk for some time. IPv6 support is added as a separate module. IPv6 EIGRP is configured and managed separately from IPv4 EIGRP, but the mechanisms and configuration techniques will be very familiar to engineers skilled with EIGRP for IPv4.

For example, both the IPv4 and IPv6 EIGRP implementations include a “shutdown” feature, which allows the routing protocol to be configured but easily disabled. Both use the Diffusing Update Algorithm (DUAL) to optimize the routing path. Both are scalable to large networks. There are also a few differences in the IPv4 and IPv6 features. For example, by contrast with IPv4 EIGRP, IPv6 EIGRP is configured over a link—there is no “network” statement as there is for IPv4.



The basic components of EIGRP for IPv6 remain the same as the IPv4 version.

EIGRP uses a small hello pack to discover other EIGRP-capable routers on directly attached links and forms durable neighbor relationships. Updates may be acknowledged using a reliable transport protocol, or they may be unacknowledged—depending on the specific function being communicated. The protocol provides the flexibility needed to unicast or multicast updates, acknowledged or unacknowledged.

Hello packets and updates are set to the well-known, link-local multicast address FF02::A, which Cisco has obtained from Internet Assigned Numbers Authority (IANA). This multicast distribution technique is more efficient than the broadcast mechanism used by earlier, more primitive routing protocols such as RIP version 1 (RIPv1). EIGRP for IPv4 also uses multicast for update distribution.

Tip	EIGRP incorporates some features that resemble features found in link-state routing protocols. For example, EIGRP sends hello messages and forms neighbor relationships with other EIGRP-capable routers. It also sends incremental routing updates to neighbors, like a link-state protocol, rather than a periodic complete update.

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.

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.

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.

EIGRP updates contain five metrics: minimum bandwidth, delay, load, reliability, and maximum transmission unit (MTU). Of these five metrics, by default, only minimum bandwidth and delay are used to compute best path. Unlike most metrics, minimum bandwidth is set to the minimum bandwidth of the entire path, and it does not reflect how many hops or low bandwidth links are in the path. Delay is a cumulative value which increases by the delay value of each segment in the path.

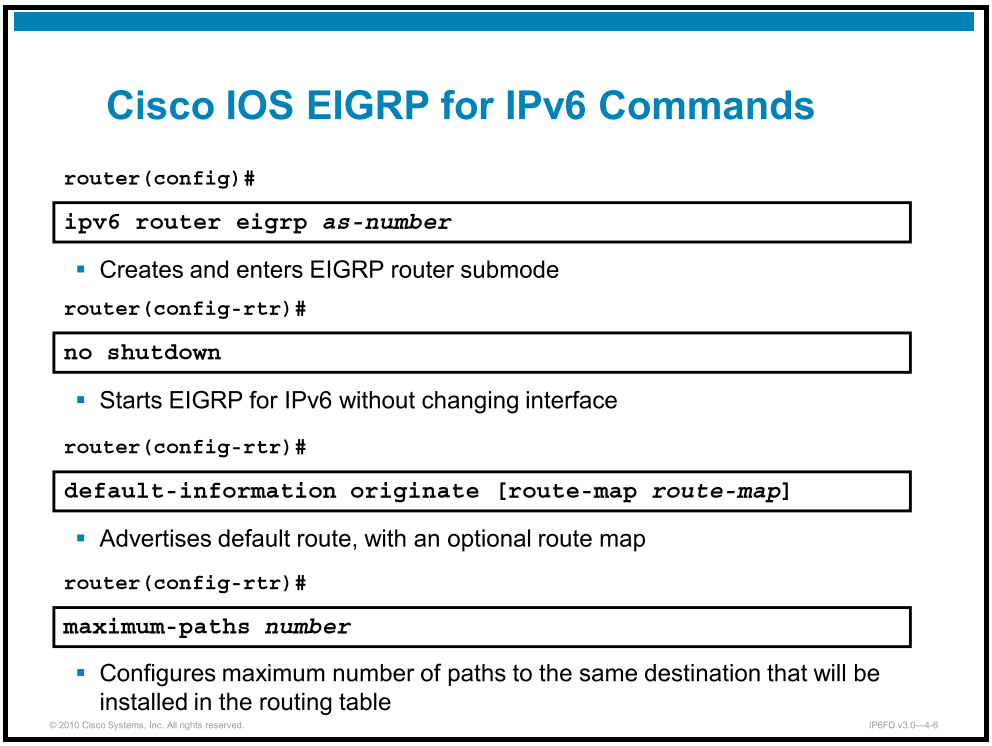
When a router discovers a new neighbor, it records the neighbor's address and interface as an entry in the neighbor table. One neighbor table exists for each protocol-dependent module. When a neighbor sends a hello packet, it advertises a hold time, which is the amount of time that a router treats a neighbor as reachable and operational. If a hello packet is not received within the hold time, the hold time expires and DUAL is informed of the topology change.

The neighbor-table entry also includes information required by RTP. Sequence numbers are employed to match acknowledgments with data packets, and the last sequence number received from the neighbor is recorded so that out-of-order packets can be detected. A transmission list is used to queue packets for possible retransmission on a per-neighbor basis. Round-trip timers are kept in the neighbor-table entry to estimate an optimal retransmission interval.

The topology table contains all destinations advertised by neighboring routers. The protocol-dependent modules populate the table, and the table is acted on by the DUAL finite-state machine. Each entry in the topology table includes the destination address and a list of neighbors that have advertised the destination. For each neighbor, the entry records the advertised metric, which the neighbor stores in its routing table. An important rule that distance vector protocols must follow is that if the neighbor advertises this destination, it must use the route to forward packets.

Cisco IOS EIGRP for IPv6 Commands

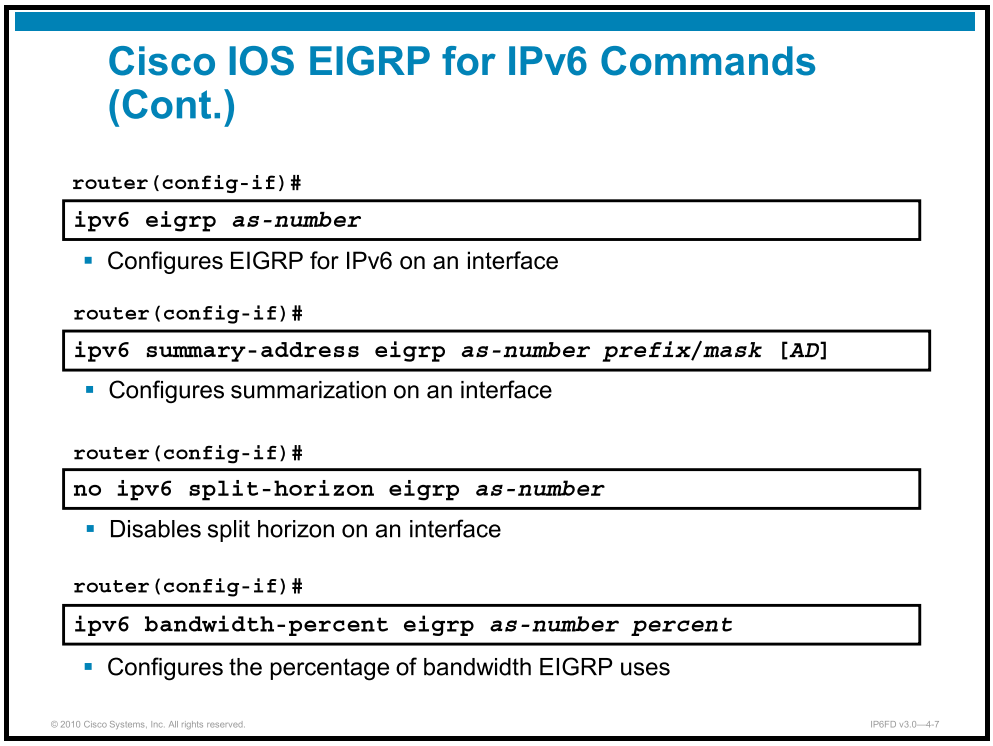
This topic describes how to configure EIGRP for IPv6 on Cisco routers.



The table lists some common configuration commands for EIGRP for IPv6. The syntax for these commands is similar, if not identical, to their IPv4 counterparts.

Cisco IOS EIGRP for IPv6 Commands

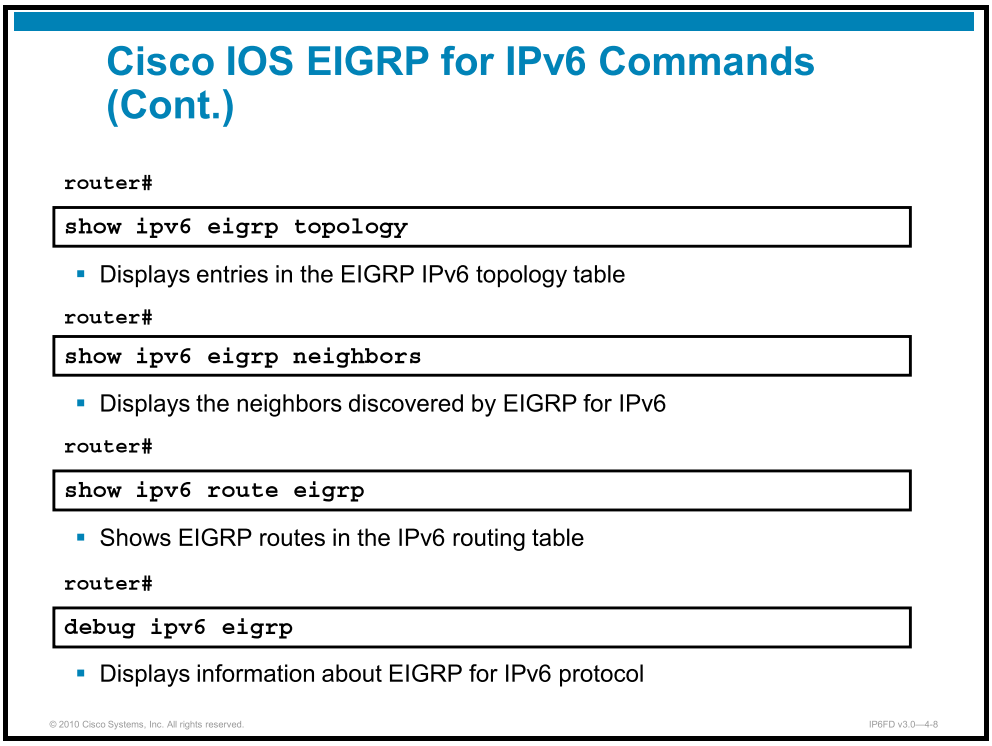
Command	Description
ipv6 router eigrp *as-number*	Enters router configuration mode and creates an EIGRP IPv6 routing process.
no shutdown	EIGRP for IPv6 has a shutdown feature. The routing process should be in "no shut" mode in order to start running.
ipv6 eigrp *as-number*	Enables EIGRP for IPv6 on a specified interface.
ipv6 bandwidth-percent eigrp *as-number percent*	Configures the percentage of bandwidth that may be used by EIGRP for IPv6 on a specified interface.



The table lists some more common configuration commands for EIGRP for IPv6. The syntax for these commands is similar, if not identical, to their IPv4 counterparts.

Cisco IOS EIGRP for IPv6 Commands

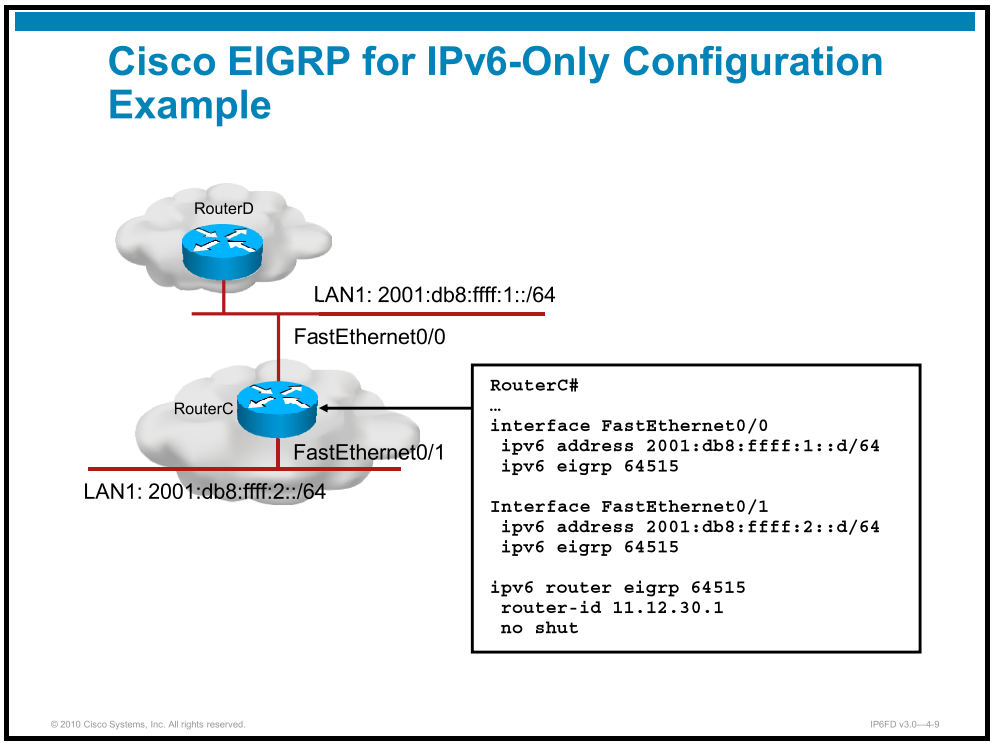
Command	Description
ipv6 eigrp *as-number*	Enables EIGRP for IPv6 on a specified interface.
ipv6 summary-address eigrp ***as-number ipv6-address*** **[*admin-distance*]**	Configures a summary aggregate address for a specified interface.
no ipv6 split-horizon eigrp *as-number*	Disables split horizon rule on a specified interface.
ipv6 bandwidth-percent eigrp *as-number percent*	Configures the percentage of bandwidth that may be used by EIGRP for IPv6 on a specified interface.



The table lists some EIGRP for IPv6 **show** and **debug** commands. The syntax for these commands is similar, if not identical, to their IPv4 counterparts.

Cisco IOS EIGRP for IPv6 show and debug Commands

Command	Description
`show ipv6 eigrp topology`	Displays entries in the EIGRP IPv6 topology table.
`show ipv6 eigrp neighbors`	Displays the neighbors discovered by EIGRP for IPv6.
`show ipv6 route eigrp`	Shows EIGRP routes in the IPv6 routing table.
`debug ipv6 eigrp`	Starts debugging for EIGRP for IPv6.

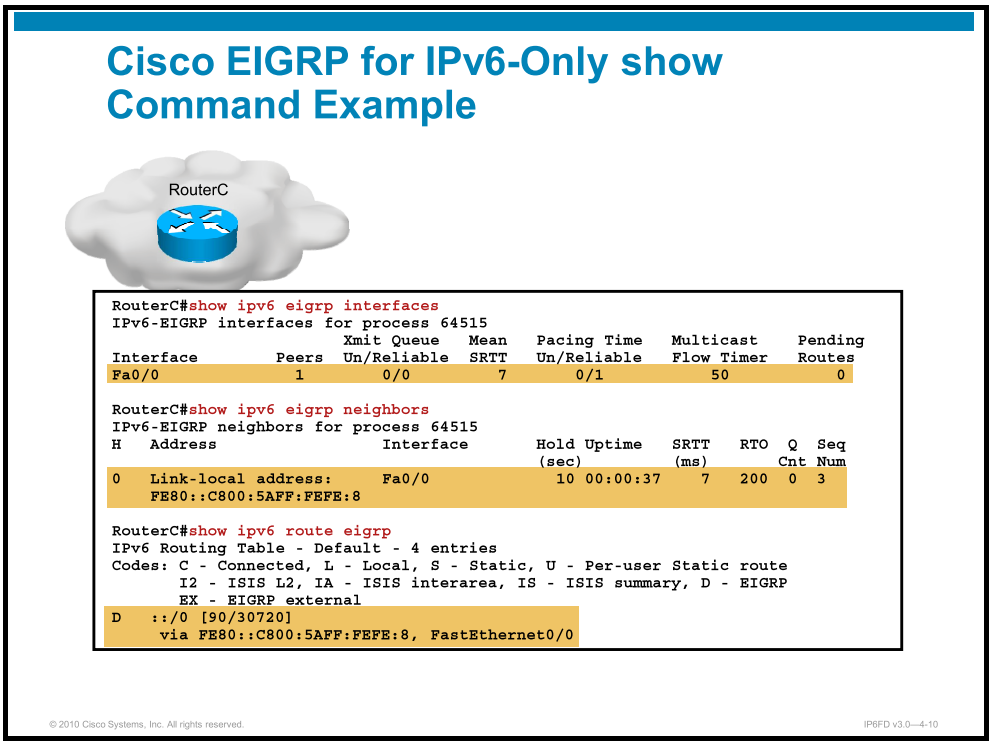


The example is a two-router network in which the routers are configured as IPv6-only EIGRP routers. Notice that you have to specify a 32-bit router ID.

Also note that the protocol is created in a “shut” state and must be in a “no shut” state to start the protocol. Many implementations of EIGRP use distribute lists. Consequently, it is not recommended to start EIGRP for IPv6 until the distribute list has been parsed, which occurs in router mode. Therefore, EIGRP for IPv6 starts in the default state of “shut.”

EIGRP multiprotocol creates multiple routing tables for each protocol (IPv4, IPv6, IPX, and so on).

EIGRP for IPv6 is supported in Cisco IOS Release 12.4(6)T.



The figure displays three examples of **show** commands.

In the first output of the **show ipv6 eigrp interfaces** we can see one neighbor on the FastEthernet 0/0 interface, which is the only interface included in the EIGRP process.

```
RouterC#show ipv6 eigrp interfaces
IPv6-EIGRP interfaces for process 64515
                        Xmit Queue    Mean    Pacing Time    Multicast    Pending
Interface         Peers  Un/Reliable   SRTT    Un/Reliable    Flow Timer   Routes
Fa0/0               1        0/0          7       0/1            50           0
```

The second example shows the output of **show ipv6 eigrp neighbos** command. Details for the neighbor are displayed, such as link-local address, interface, hold time and uptime.

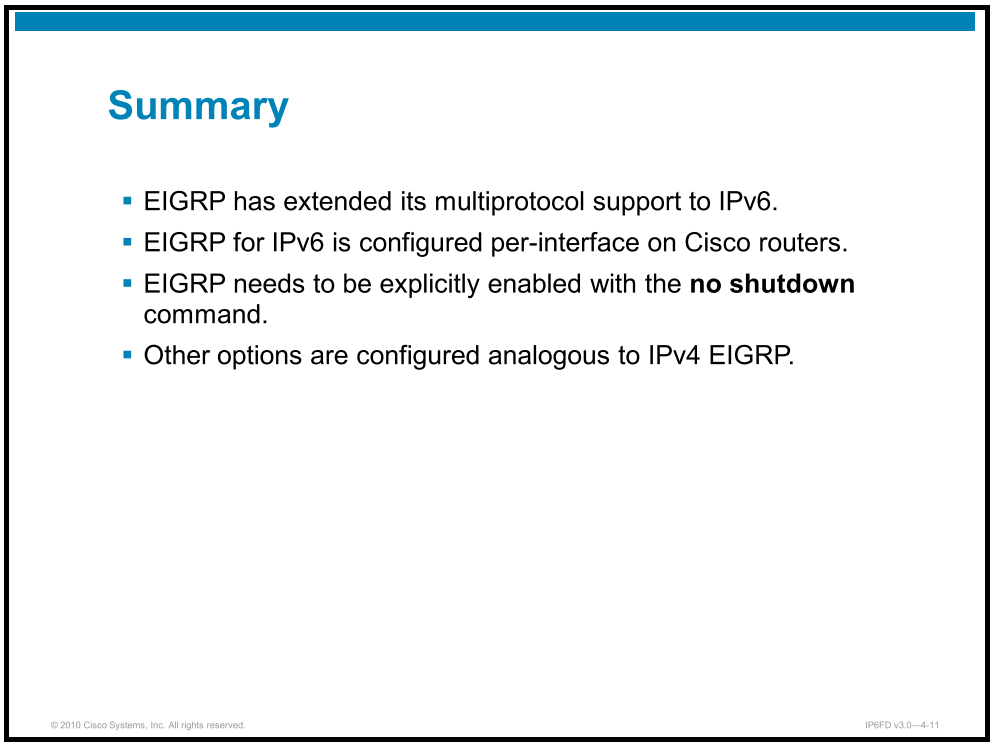
```
RouterC#show ipv6 eigrp neighbors
IPv6-EIGRP neighbors for process 64515
H   Address                 Interface       Hold Uptime    SRTT    RTO  Q  Seq
                                            (sec)          (ms)       Cnt Num
0   Link-local address:     Fa0/0             10 00:00:37    7    200  0  3
    FE80::C800:5AFF:FEFE:8
```

The third example shows the output of **show ipv6 route eigrp** command. Here we are presented with a default route, learned by the EIGRP routing protocol.

```
RouterC#show ipv6 route eigrp
IPv6 Routing Table - Default - 4 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
D   ::/0 [90/30720]
     via FE80::C800:5AFF:FEFE:8, FastEthernet0/0
```

Summary

This topic summarizes the key points discussed in this lesson.



Lesson 5

Understanding MP-BGP

Overview

The global Internet routing infrastructure is largely built using Border Gateway Protocol (BGP). Organizations wishing to announce presence and reachability on the Internet need listings in the Default Free Zone BGP routing tables. The IP version 6 (IPv6) Internet continues to use BGP, which needed modifications to support it. This lesson covers Multiprotocol Border Gateway Protocol (MP-BGP), including operation, IPv6-related configuration, and commands.

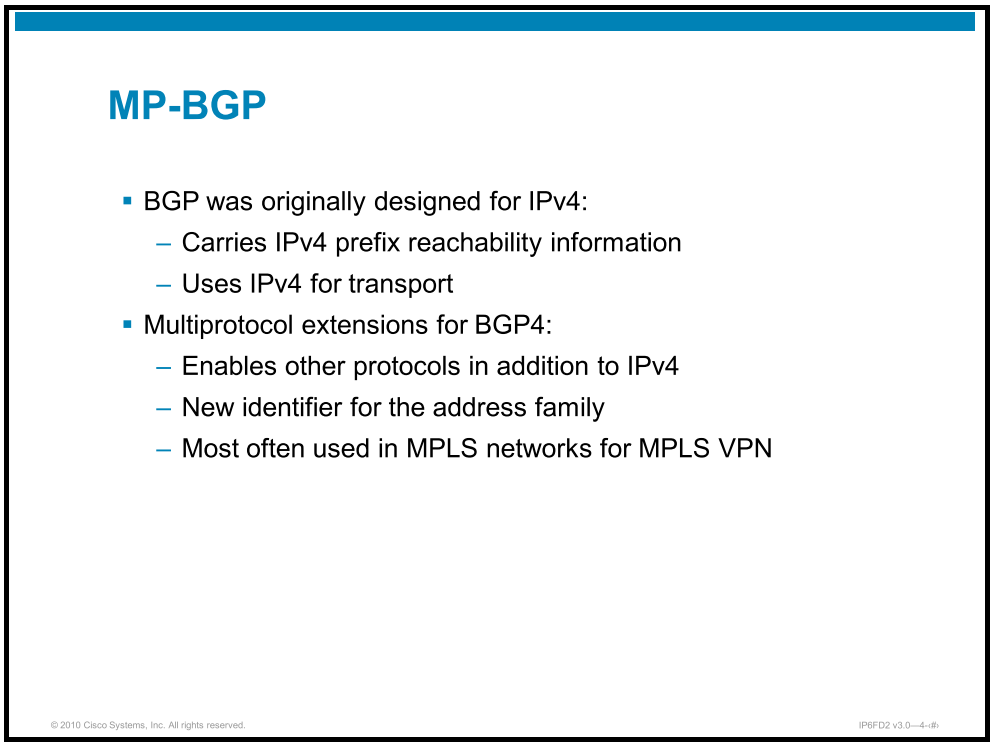
Objectives

Upon completing this lesson, you will be able to describe how MP-BGP supports IPv6 and how to configure MP-BGP on Cisco routers. This ability includes being able to meet these objectives:

- Describe how the MP-BGP routing protocol is supported in IPv6
- Describe the changes made to MP-BGP to support IPv6
- Examine BGP peering over link-local addresses
- Examine BGP prefix filtering
- Configure and troubleshoot the MP-BGP protocol on Cisco routers

MP-BGP Support for IPv6

This topic describes the extensions made to MP-BGP to support IPv6.



Original BGP-4 (RFC 1771) carries IPv4 specific information. Multiprotocol BGP (RFC 2283) adds capability to transport routing information for other protocols by using extensions. Multiprotocol extensions for Border Gateway Protocol (BGP) are defined as new attributes. These new attributes define Network Layer Reachability Information (NLRI) and a next hop (the next router in the path to the destination) where IP version 6 (IPv6) addresses can be used.

Note	The next-hop must be of the same address type (address family) as the NLRI exchanged. An IPv6 route cannot have an IPv4 next hop address.

BGP4 with multiprotocol extensions enables the use of many address families. Address families define the type of addresses being carried. The most common address families are IPv4, IPv6, and VPNv4 and VPNv6 for MPLS VPN routes.

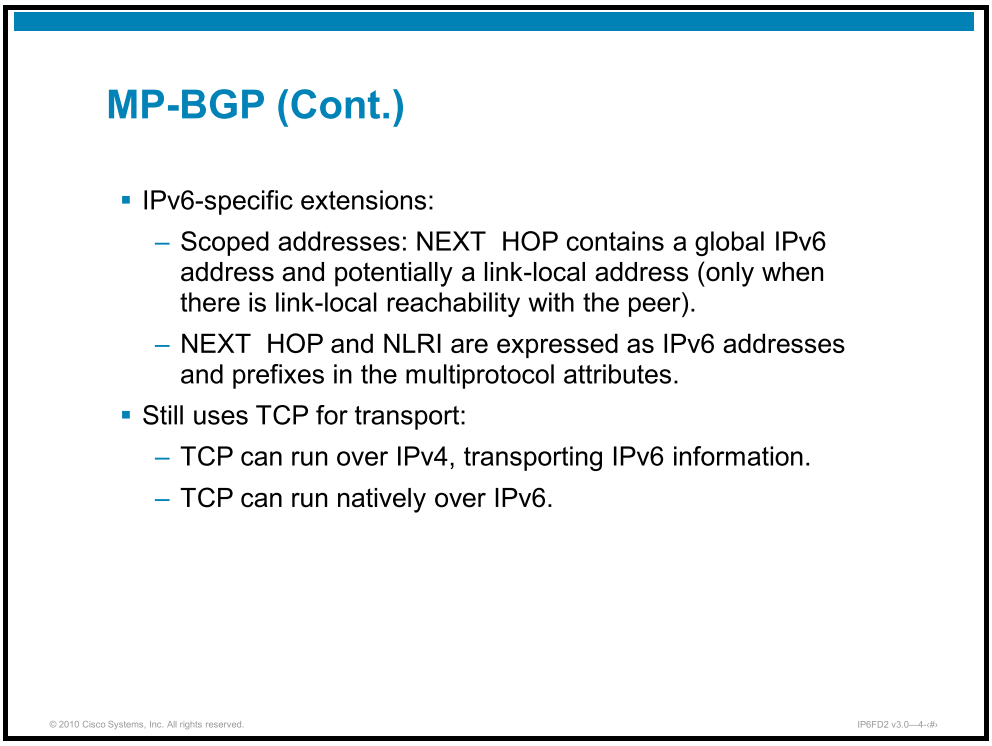
An address family is activated within BGP using the **address-family** command in BGP routing protocol configuration (router configuration mode). Afterwards, an IPv6 neighbor needs to be activated within that address family using the **neighbor activate** command.

Two new attributes support multiprotocol BGP (MP-BGP):

- Multiprotocol reachable NLRI (MP_REACH_NLRI)
- Multiprotocol unreachable NLRI (MP_UNREACH_NLRI)

The MP_REACH_NLRI attribute describes reachable destinations and includes attributes that contain information about network-layer protocol prefixes (specifically IPv6), and the next destination hop-to-reach prefixes.

Note	MP-BGP is defined in RFC 2858, *Multiprotocol Extensions for BGP4*.



IPv6-specific extensions for MP-BGPv4 are as follows:

- **Scoped addresses:** NEXT HOP contains a global IPv6 address, or potentially a link-local address, when there is link-local reachability with the peer.

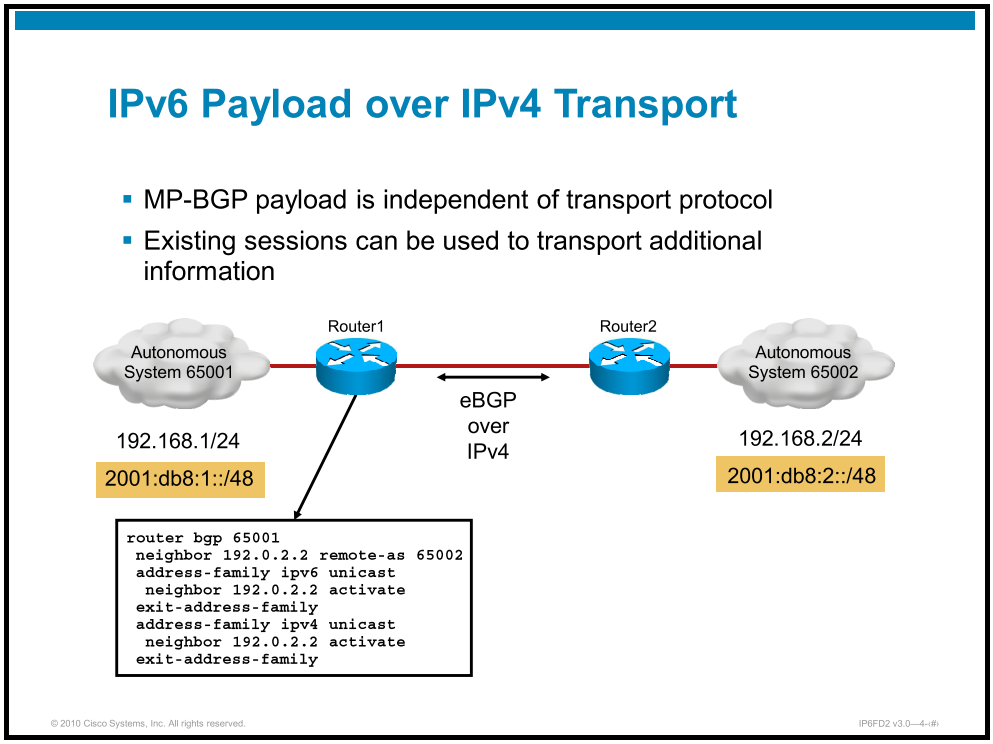
Note	When link-local addresses are used for peering with a neighbor BGP router, these link-local IPv6 addresses are used as next hop IP addresses for the routes carried by BGP. In the majority of cases, the next hop IPv6 addresses need to be changed to global IPv6 by attaching a route map to the **neighbor** configuration statement.

- **IPv6 address format:** NEXT HOP and NLRI are expressed as IPv6 addresses and prefixes in the multiprotocol attributes.

Note	You can run MP-BGP over IPv4 or IPv6 transport and can exchange routes for IPv4, IPv6, or both. BGP uses the TCP protocol for peering and this has no relevance to the routes carried inside the BGP exchanges. Both IPv4 or IPv6 can be used to transport a TCP connection on the network layer.

IPv6 as Payload and Transport Mechanism in MP-BGP

This topic describes the changes made to MP-BGP to support IPv6.

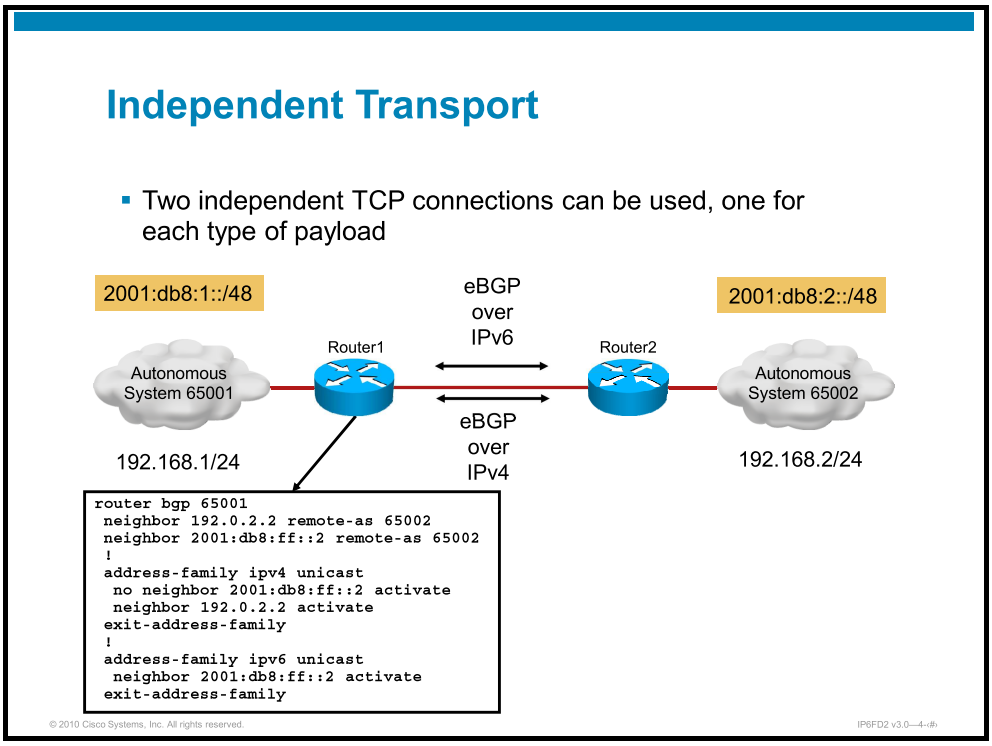


MP-BGP routing information is not related to transport session. This means that existing IPv4 TCP sessions can be upgraded to also carry IPv6 routing information when adding IPv6 support to networks.

An existing neighbor can be activated for the IPv6 address family and IPv6 routing information will be sent over the same neighbor session.

Note	However, when configuring an additional address family (IPv6) for that particular neighbor, the routers will start again the capability exchange negotiation for that session by resetting it. In the network, this will be noticed as a route flap which is undesirable.

Additionally, modification of the next hop attribute is necessary, as by default BGP uses neighbor IP address for the next hop. Since there is an IPv4 session established to carry IPv6 routes, an IPv4 address will appear as the next hop IP address for IPv6 prefixes. This needs to be corrected manually by configuring and attaching a route map to the **neighbor** configuration statement. The route map should set an IPv6 next hop IP address to IPv6 prefixes, and this next hop IPv6 address must be reachable either globally and configured on the link, or reachable by the underlying IGP.



When complete separation of IPv4 and IPv6 is required, native connection can be used. Also, if an IPv6 only router is deployed, typically it will not carry any IPv4 information, nor will it have IPv4 addresses on its interfaces.

```
router bgp 65001
 neighbor 192.0.2.2 remote-as 65002
 neighbor 2001:db8:ff::2 remote-as 65002
 !
 address-family ipv4 unicast
  no neighbor 2001:db8:ff::2 activate
  neighbor 192.0.2.2 activate
 exit-address-family
 !
 address-family ipv6 unicast
  neighbor 2001:db8:ff::2 activate
 exit-address-family
```

In the example in the figure, two BGP sessions are established between two routers. One session is configured to run on IPv4 and it carries only IPv4 routing information.

The second session is established using TCP over IPv6 and is used only for IPv6 routing.

The IPv6 neighbor must be explicitly activated in IPv6 address family, because IPv6 address family is not enabled by default. IPv4 address family is enabled for all neighbors by default, so you must explicitly disable it by using the **no neighbor activate** command.

The scenario when using separate sessions for IPv4 and IPv6 prefix exchange does not create a route flap when IPv6 support is being added to a BGP router which was running previously only IPv4. This method is preferred, as it does not require to change the next hop attribute for the prefixes because the router uses the neighbor IPv6 address. The link between the routers must be dual-stacked.

Pros and Cons

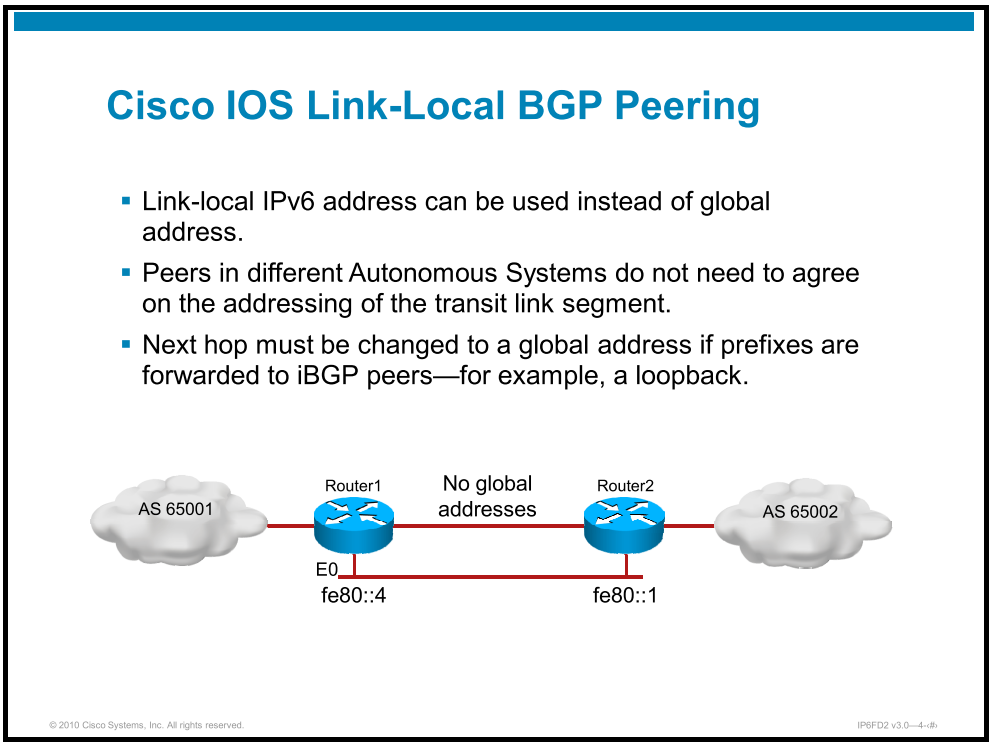
- Using a single TCP session:
  - Reduces number of neighbors
  - IPv6 over IPv4 session requires modification of next hop attribute
  - Only one session must be reset every time filters are updated
  - Same route reflector setup can be used
  - IPv6 influences IPv4 BGP environment (potential risk)
- Using separate TCP sessions:
  - Complete independence of IPv4 and IPv6
  - Additional neighbor configuration required
  - Next hop can be taken from neighbor address
  - Non-IPv4 neighbors are not seen in the **show ip bgp summary** output

Decision whether to combine both address families on one neighbor or separate sessions lies with the network administrator. There are advantages to both approaches.

Using a single neighbor can reduce number of neighbor sessions. In an environment where a lot of neighbors are configured this can significantly reduce size and complexity of configuration.

BGP Peering Over Link-Local Addresses

This topic describes how to examine BGP peering over link-local addresses.

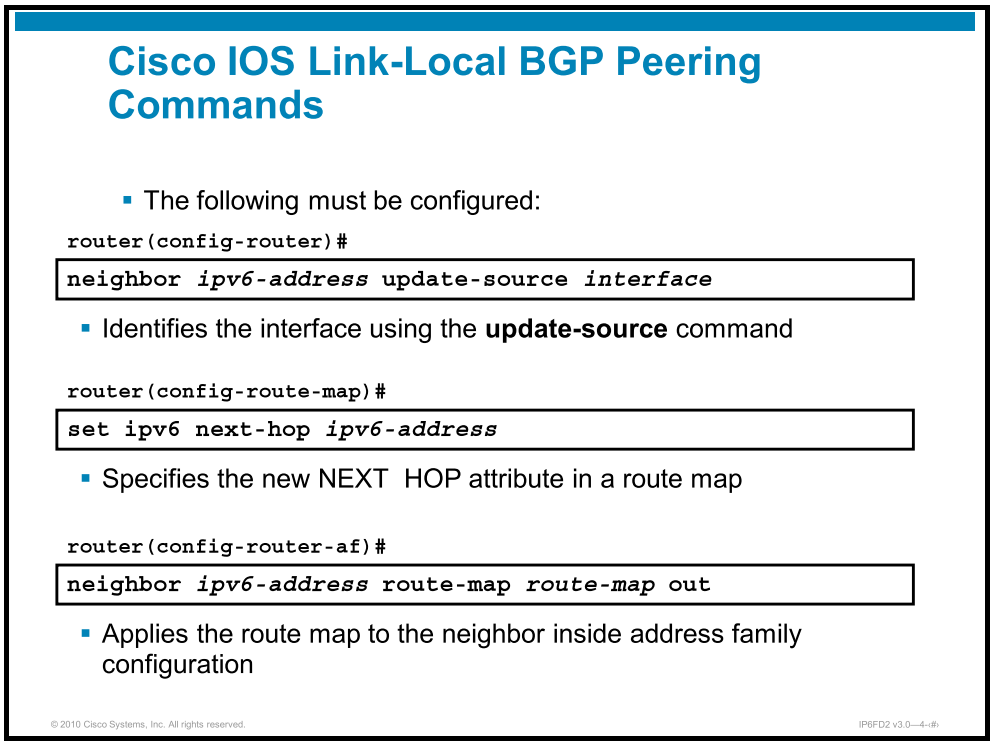


Either Internal Border Gateway Protocol (IBGP) or External Border Gateway Protocol (EBGP) peering can occur using the link-local address of the neighbor rather than the global address of the neighbor. Using link-local IPv6 addresses for Border Gateway Protocol (BGP) peering is useful because no IPv6 address allocation is necessary for the link between the BGP peers.

When specifying a link-local address for peering, you must identify the interface associated with that link-local address by using the **neighbor update-source** command. The router has no mechanism to know which link-local address to use if more than one IPv6 interface is configured. Without the **neighbor update-source** command, the TCP session between the BGP routers cannot be established.

When using link-local addresses for peering, you must use a route map to set the next-hop attribute on outbound BGP update messages to the neighbor. This route map will set the next-hop attribute to both the link-local and the global IPv6 addresses of the identified interface. If the route map is not set, the BGP next-hop attribute will be set to ::, and the update messages will be ignored by the neighbor.

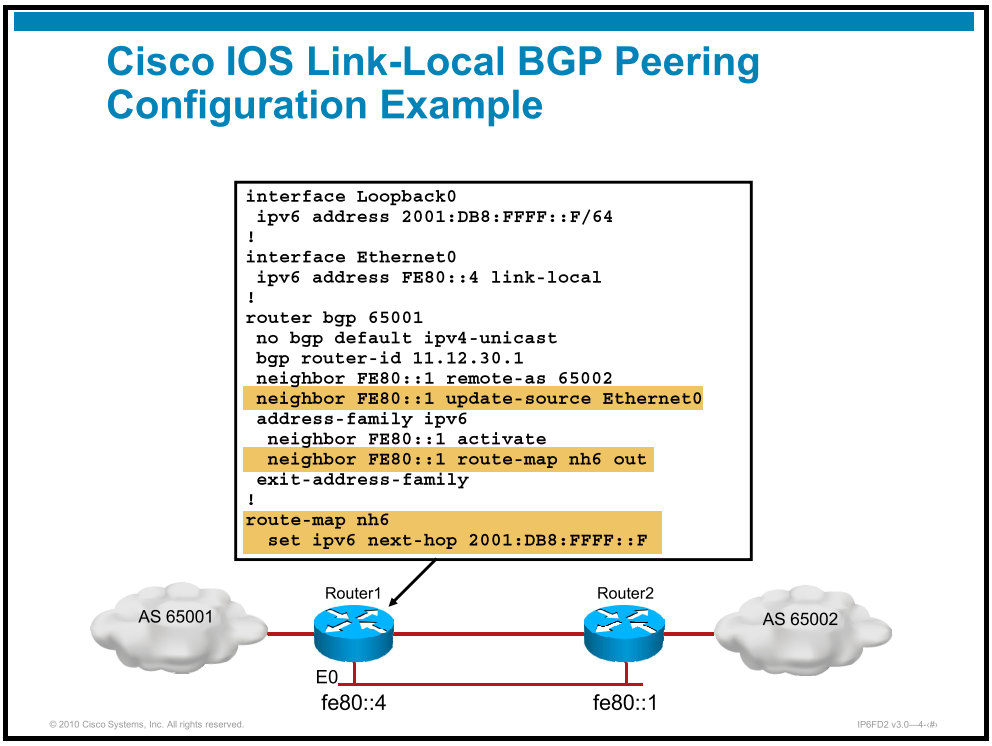
Tip Using link-local addresses for BGP peering is most commonly seen at interexchange points. These points are where ISPs and other large organizations meet at a collocation facility, and each puts a router on a common Layer 2 subnet. In this case, using link-local addresses is advantageous because no global-scope addresses need to be used on the peering subnet. Given the large address space in IPv6, prefix conservation is not the main motivator here, rather it is an effort by two parties to have a “neutral meet,” with a clean demarcation between their routable address spaces.



The table lists required additional commands and their syntax.

Cisco IOS Link-Local BGP Peering Commands

Command	Description
neighbor *ipv6-link-local-address* update-source *interface*	For the link local neighbor you must specify outgoing physical interface.
set ipv6 next-hop *ipv6-global-address*	This command must be entered in a route map, to set the next hop to a global and reachable IPv6 address. This will usually be an address on a loopback interface.
neighbor ***ipv6-link-local-address*** **route-map *route-map* out**	Finally, the route map which alters the next hop attribute must be applied to the neighbor in the outbound direction.



In the example, two routers from different autonomous systems are peering using the link-local IPv6 address. The **neighbor fe80::1 remote-as 65002** command identifies the peer with its link-local address and AS number. The command **neighbor fe80::1 update-source Ethernet0** specifies the outgoing interface that is used for the peering.

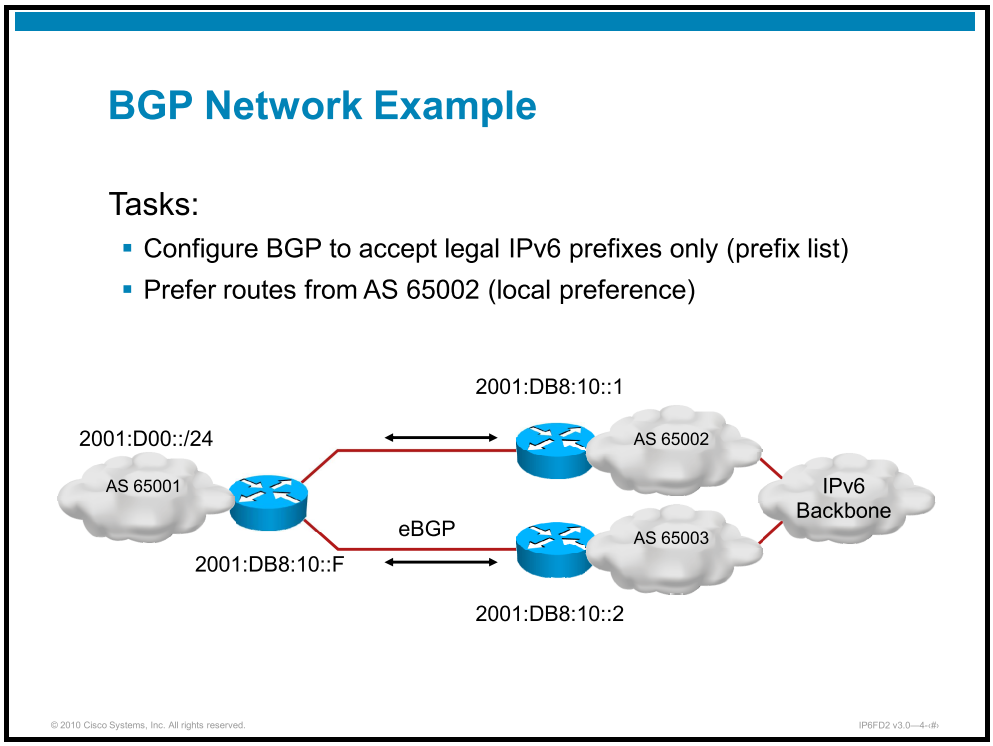
The **neighbor fe80::1 route-map nh6 out** command applies the route map nh6 on outbound updates. The **route-map nh6** command sets the next-hop attribute to the global IPv6 address.

The example shows that a link-local address is manually configured on the Ethernet0 interface. It is also possible to let the router automatically configure this link-local address by replacing the **ipv6 address fe80::4 link-local** command with **ipv6 enable**. The latter instructs the router to assign a dynamic link-local address using the extended universal identifier 64-bit format (EUI-64) for the interface identifier.

Note EUI-64 derived IPv6 addresses are practical to configure but impose the risk that if the interface or router (or MAC address) is changed, the configuration on the peer router becomes invalid since the EUI-64 derived link-local IPv6 address will change. If link-local addresses are used, it is better that they are manually configured link-local IPv6 addresses.

BGP Prefix Filtering

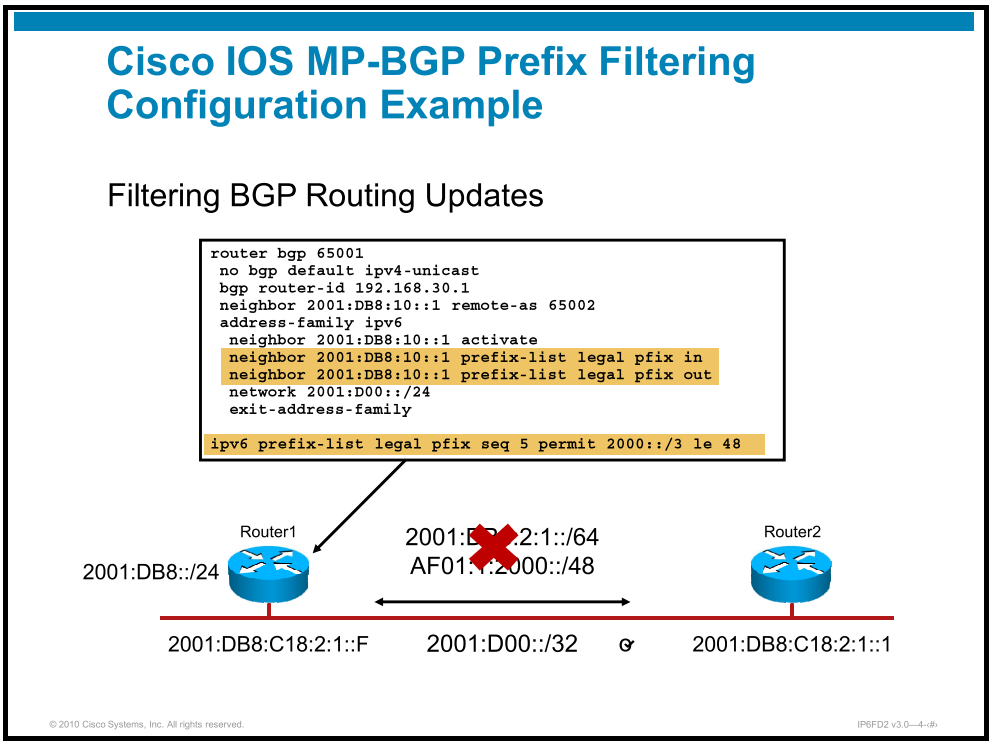
This topic describes how to examine BGP prefix filtering.



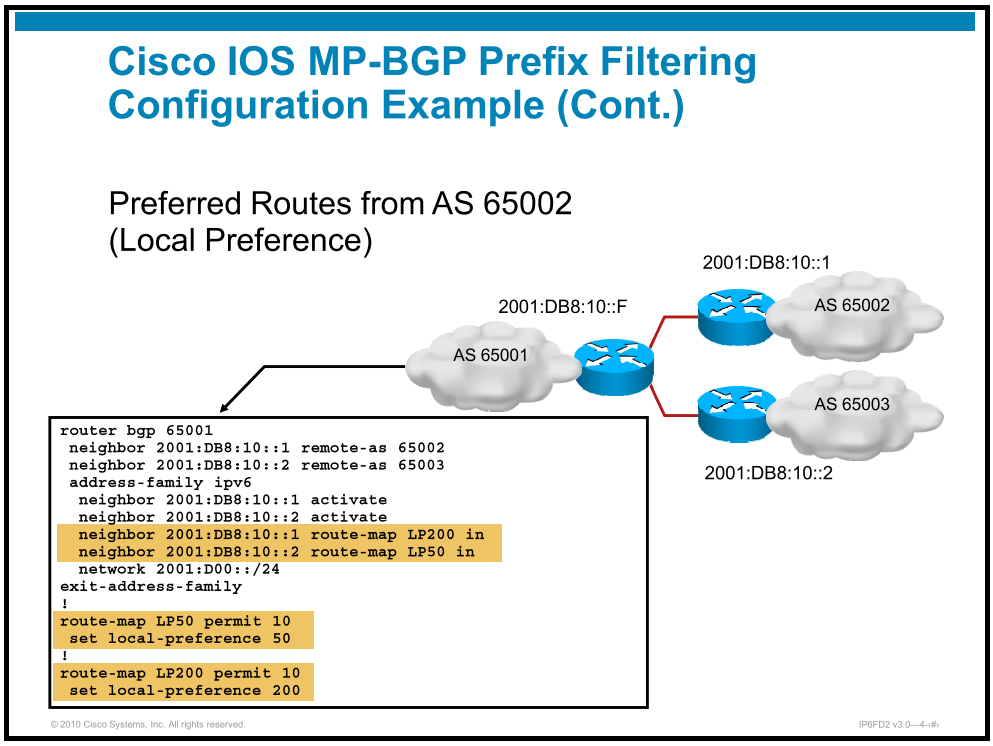
One of the strengths of BGP is vast array of facilities for route filtering and modification. Most BGP deployments will implement at least some simple route filtering, either based on AS path or network prefix.

When filtering based on network prefix, you will most often use IPv6 prefix lists to define range of addresses. All concepts of writing of IPv6 prefix lists are the same as with IPv4 prefix lists, except for the change in address format.

Following examples will show how to implement simple prefix-based filtering of routing updates and how to perform simple modification of routes using route maps.



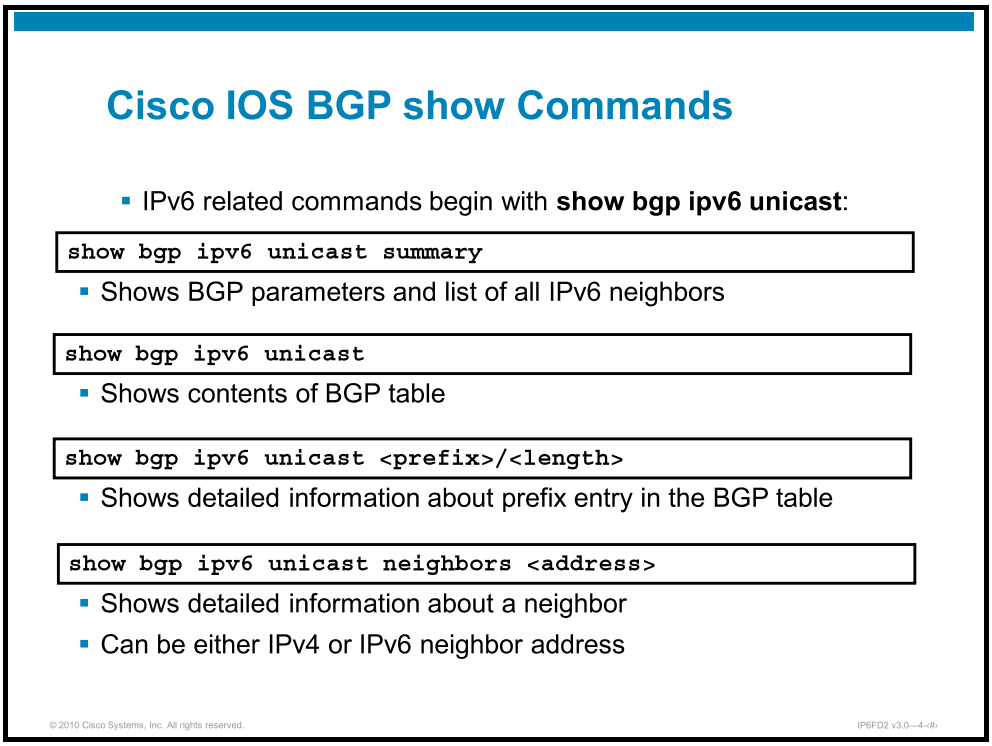
You can filter BGP routing updates on the basis of prefix information from BGP update messages. The example illustrates a BGP filter list *legal pfix* which will allow only addresses within 2000::/3 which is part of the IPv6 address space currently allocated for global addresses. The filter is applied to the inbound announcements received from the two BGP peers. The same prefix list could be applied to the outgoing announcements sent by AS 65001.



You can tune the BGP path selection by modifying the local preference on routes received from a peer. In this example, routes received from AS 65002 will have a local preference of 200 instead of the default, which is 100, while routes from autonomous system 65003 will have local preference lowered to 50. If the same route is received from both AS 65002 and AS 65003, the path to AS 65002 will be preferred.

MP-BGP Configuration and Troubleshooting

This topic describes how to configure and troubleshoot MP-BGP on Cisco routers.



show bgp ipv6 summary

The **show bgp ipv6 unicast summary** and **show bgp ipv6 multicast summary** commands provide output similar to the **show ip bgp summary** command, except they are IPv6-specific.

show bgp ipv6 {unicast | multicast} summary

show bgp ipv6

To display entries in the IPv6 Border Gateway Protocol (BGP) routing table, use the **show bgp ipv6** command in user EXEC or privileged EXEC mode. This command provides output similar to the **show ip bgp** command, except that it is IPv6-specific. Detailed output can be obtained by specifying a prefix at the end.

To display the top of the stack label with label switching information, enter the **show bgp ipv6** EXEC command with the **labels** keyword.

Note	If a prefix has not been advertised to any peer, the display shows "Not advertised to any peer."

show bgp ipv6 {unicast | multicast} [*ipv6-prefix/prefix-length***] [longer-prefixes] [labels]**

show bgp ipv6 Parameters

Parameter	Description
`unicast`	Specifies IPv6 unicast address prefixes.
`multicast`	Specifies IPv6 multicast address prefixes.

Parameter	Description
ipv6-prefix	(Optional) IPv6 network number, entered to display a particular network in the IPv6 BGP routing table. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
/prefix-length	(Optional) The length of the IPv6 prefix. 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.
`longer-prefixes`	(Optional) Displays the route and more specific routes.
`labels`	(Optional) Displays MPLS label information.

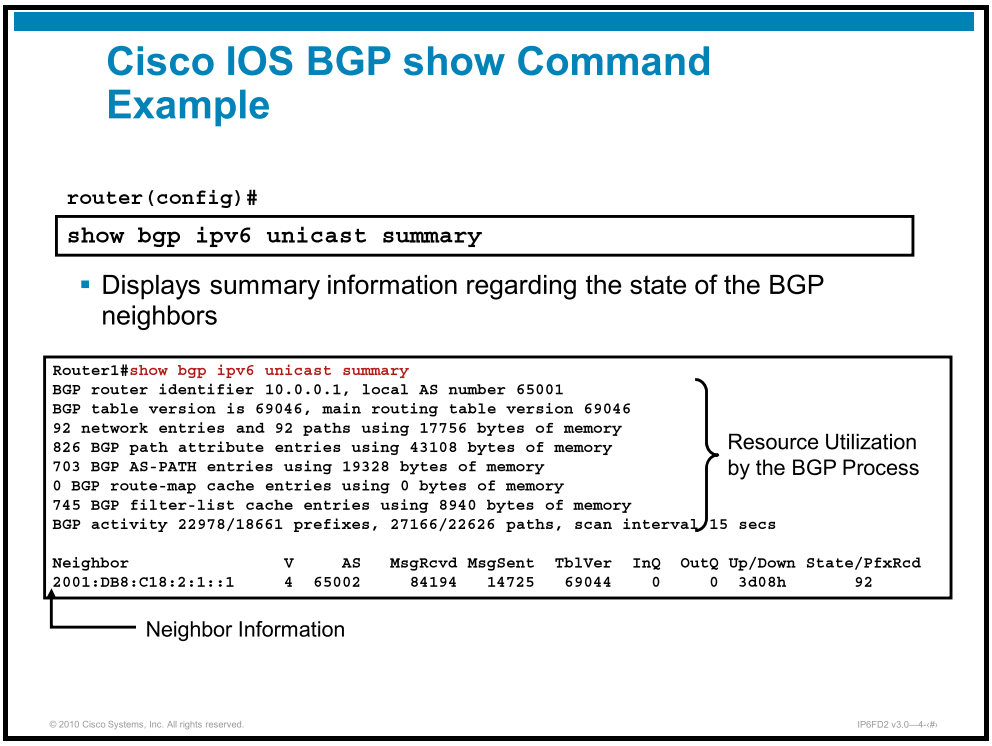
show bgp ipv6 neighbors

To display information about IPv6 Border Gateway Protocol (BGP) connections to neighbors, use the **show bgp ipv6 neighbors** command in user EXEC or privileged EXEC mode. The **show bgp ipv6 unicast neighbors** and **show bgp ipv6 multicast neighbors** commands provide output similar to the **show ip bgp neighbors** command, except they are IPv6-specific.

show bgp ipv6 {unicast | multicast} neighbors [*ipv6-address***] [received-routes | routes | flap-statistics | advertised-routes | paths** *regular-expression* **| dampened-routes]**

show bgp ipv6 neighbors Parameters

Parameter	Description
`unicast`	Specifies IPv6 unicast address prefixes.
`multicast`	Specifies IPv6 multicast address prefixes.
ipv6-address	(Optional) Address of the IPv6 BGP-speaking neighbor. If you omit this argument, all IPv6 neighbors are displayed. This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
`received-routes`	(Optional) Displays all received routes (both accepted and rejected) from the specified neighbor.
`routes`	(Optional) Displays all routes received and accepted. This is a subset of the output from the **received-routes** keyword.
`flap-statistics`	(Optional) Displays flap statistics for the routes learned from the neighbor.
`advertised-routes`	(Optional) Displays all the routes the networking device advertised to the neighbor.
`paths` *regular-expression*	(Optional) Regular expression used to match the paths received.
`dampened-routes`	(Optional) Displays the dampened routes to the neighbor at the IP address specified.



Use the **show bgp ipv6 summary** command to display information about the configured BGP neighbors. The information contains BGP neighbor activity. Examples include the number of BGP messages sent and received, the elapsed time since the BGP peering is established, and statistics on resource utilization by the local BGP process.

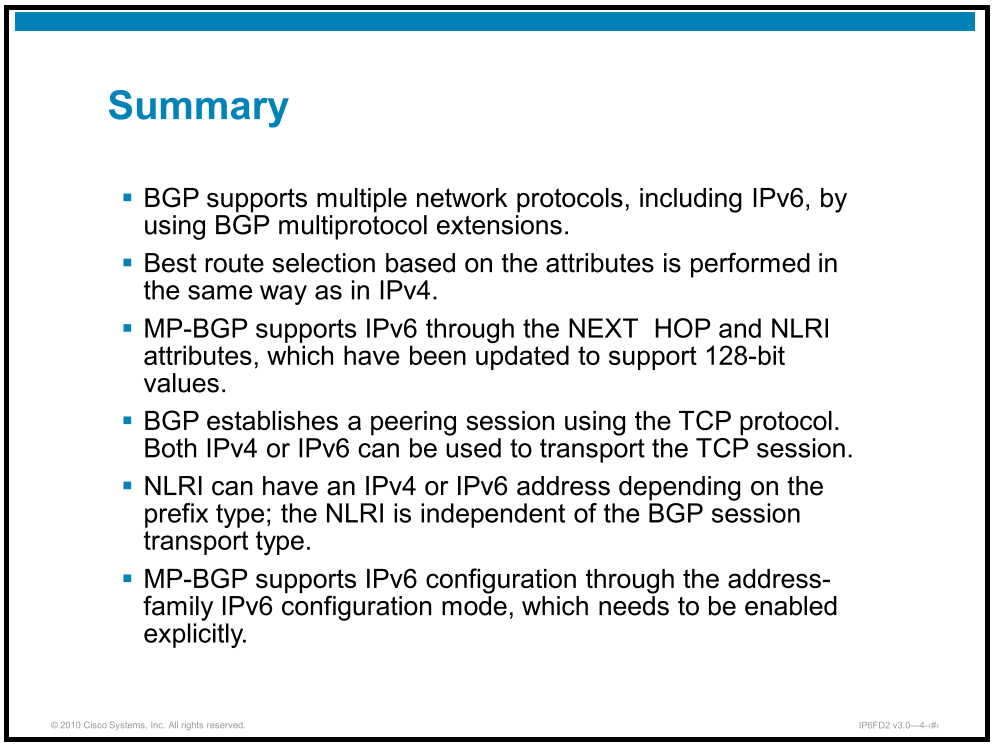
The figure shows an example of one BGP peer that was configured.

Many other **show** and **debug** commands are available, which are mostly the same as for IPv4.

Note that the equivalent IPv4 command is **show ip bgp summary**; the keywords are reversed.

Summary

This topic summarizes the key points discussed in this lesson.



Resources

To learn more about MP-BGPv4 and support for IPv6, refer to the following documents:

- *Implementing Multiprotocol BGP for IPv6*
  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-mptcl_bgp.html
- *RFC 2545: Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*
  http://www.rfc-archive.org/getrfc.php?rfc=2545

Lesson 6

Configuring IPv6 Policy-Based Routing

Overview

This lesson describes policy-based routing (PBR) for Internet Protocol version 6 (IPv6), its main usages, and how to configure it.

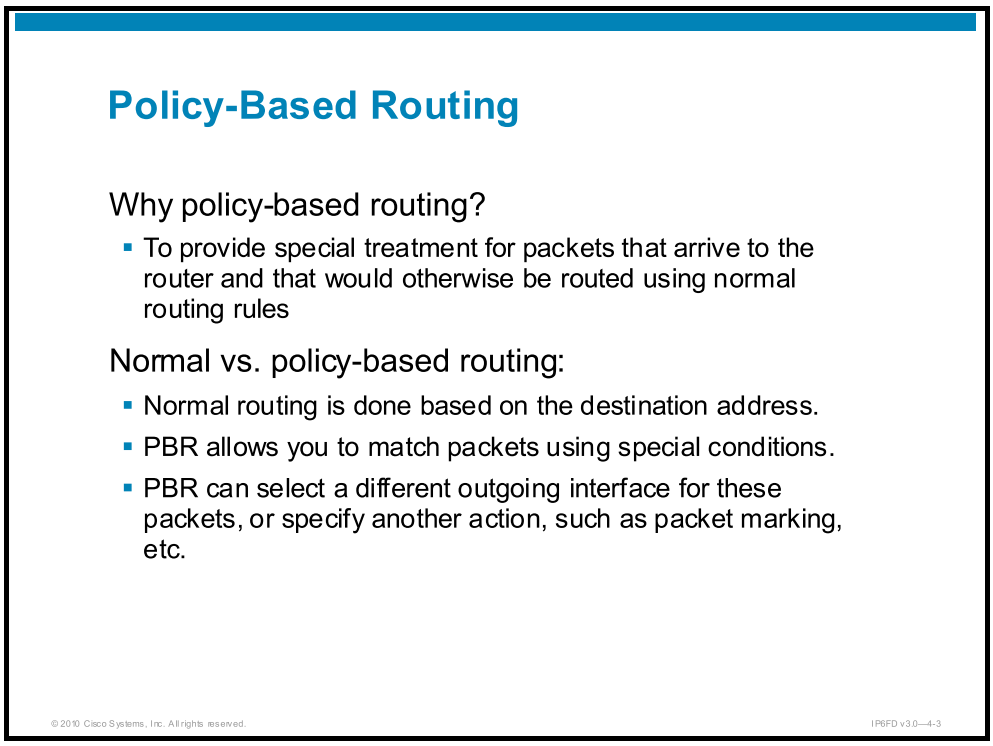
Objectives

Upon completing this lesson, you will be able to explain the issues when using PBR and when disabling the processing of extension headers. This ability includes being able to meet these objectives:

- Explain the issues when using PBR and when disabling the processing of extension headers
- Explain the configuration steps in configuring PBR

Policy-Based Routing

This topic describes how to use PBR in IPv6 and how to enable or disable processing of extension headers.



The reason to implement PBR is to provide special treatment for packets that come to the router and that would normally be routed according to the routing table but instead should be routed differently.

Normal IP (and IPv6) routing is a process that is based on the routing table of the router. When a packet comes to the router, the router performs a routing lookup and forwards the packet to the outgoing interface. The outgoing interface goes to the best path, selected by the routing protocol, or, in the case of multiple paths with equal cost, the router load-balances the packets between these paths.

There are few options to send the packets to a path that you know is available but is not the best path and therefore is not in the routing table. PBR allows you to administratively assign an outgoing interface (or IP or IPv6 next hop) for this traffic, using the standard PBR mechanisms.

Packets that should be policy-routed can be matched using several criteria that are supported in route maps, and several actions can be specified for this traffic, as follows:

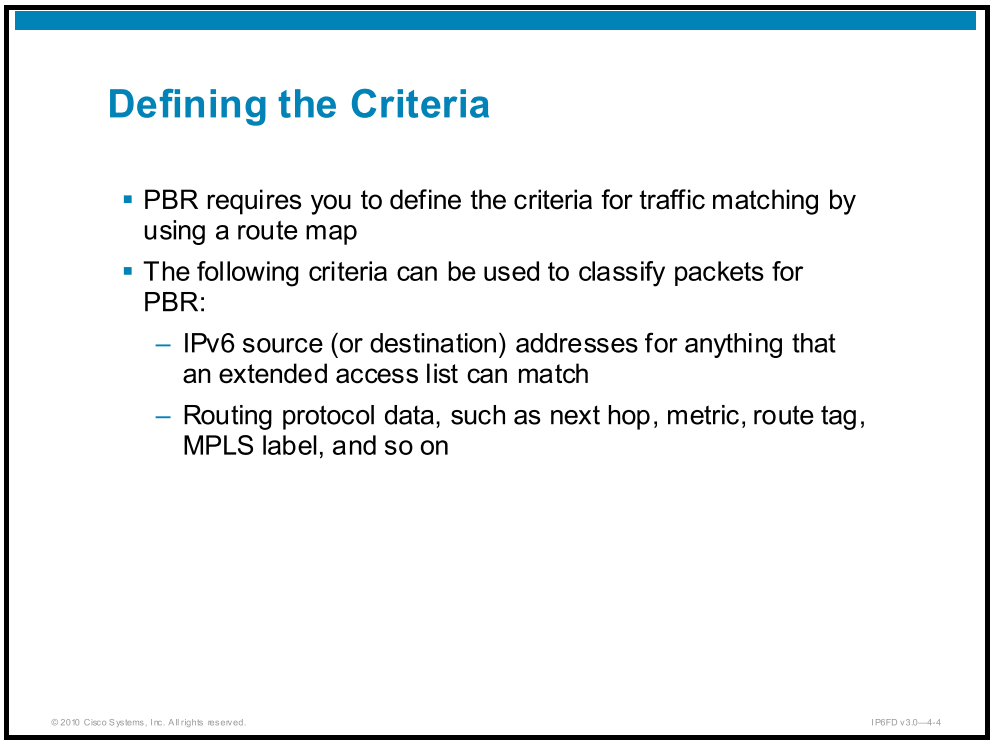
- Defining the outgoing interface
- Defining the IP or IPv6 next hop, for example, for Border Gateway Protocol-routed (BGP-routed) traffic
- Marking the packet with IP precedence to impose quality of service (QoS) and traffic engineering

Note QoS and traffic engineering require other mechanisms to be enabled on the router, such as low latency queuing (LLQ). Traffic engineering should not be confused with Cisco Multiprotocol Label Switching Traffic Engineering (MPLS TE), which is an advanced and dynamic traffic engineering solution.

To summarize, routing operations can be any of the following three types:

- **Destination-based:** The majority of routing operations in networks
- **Policy-based:** To route on anything else in addition to the destination (or destination with additional criteria), that is, PBR
- **Source-based:** To route traffic that is based on source address, which is used in multicast networks

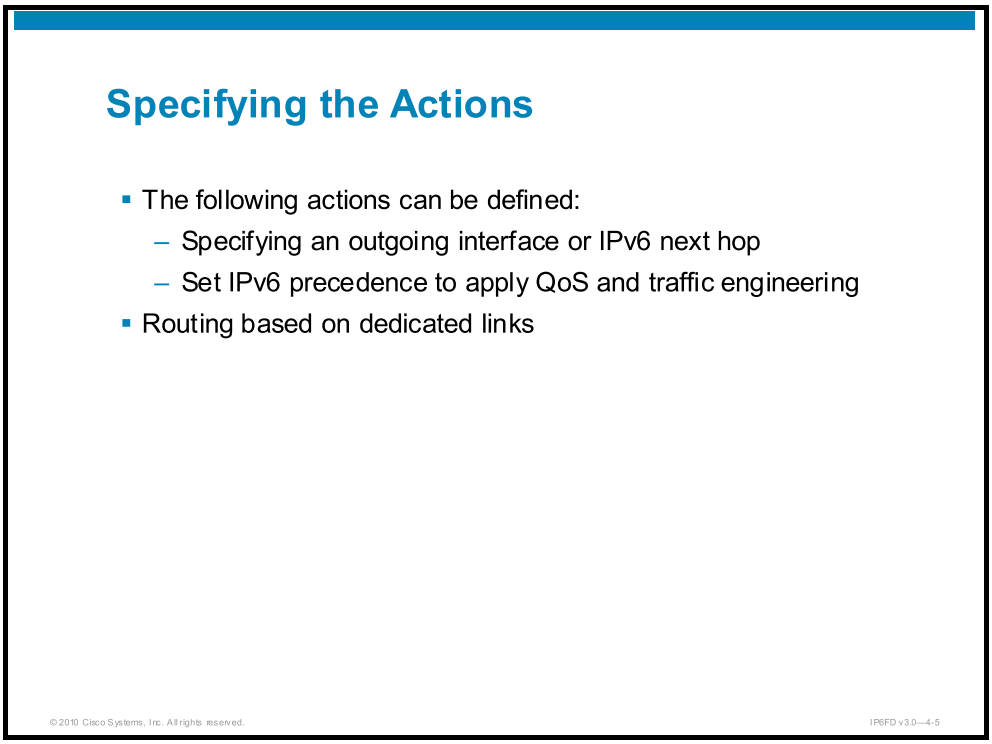
PBR gives you more control over routing by extending and complementing the existing mechanisms that are provided by routing protocols. Limited packet marking features are also available, in which PBR allows you to set the IPv6 precedence and allows you to specify a path for certain traffic, such as priority traffic over a high-cost link.



To be able to route based on policy, you should define the criteria on which packets such a policy should be applied. The Cisco IOS Software allows you to define a route map, which is a universal tool to match packets on certain conditions and set actions accordingly.

Note	In addition to PBR, route maps are used extensively when performing route redistribution from one routing protocol to another. These route maps allow complete control of conditions and actions.

Route maps consist of match and set clauses that are executed in the order that is defined in the route map. Match command sentences look for specific conditions, such as source IPv6 address, and other elements that can be matched by an extended access list (for example, address ranges, Layer 4 port numbers, header options, and so on).



Set clauses in the route map define the actions that a router should perform on the packets. The following examples are in use with PBR:

- Set the outgoing interface that the packet should use to exit the router (most common).
- Set the IP precedence marker to enforce QoS at a later point on the router or in the network.

Using a combination of matching on the source or destination IPv6 address and setting the outgoing interface, you can route certain traffic over dedicated links, even if the outgoing interface for the destination network in the routing table is a different interface.

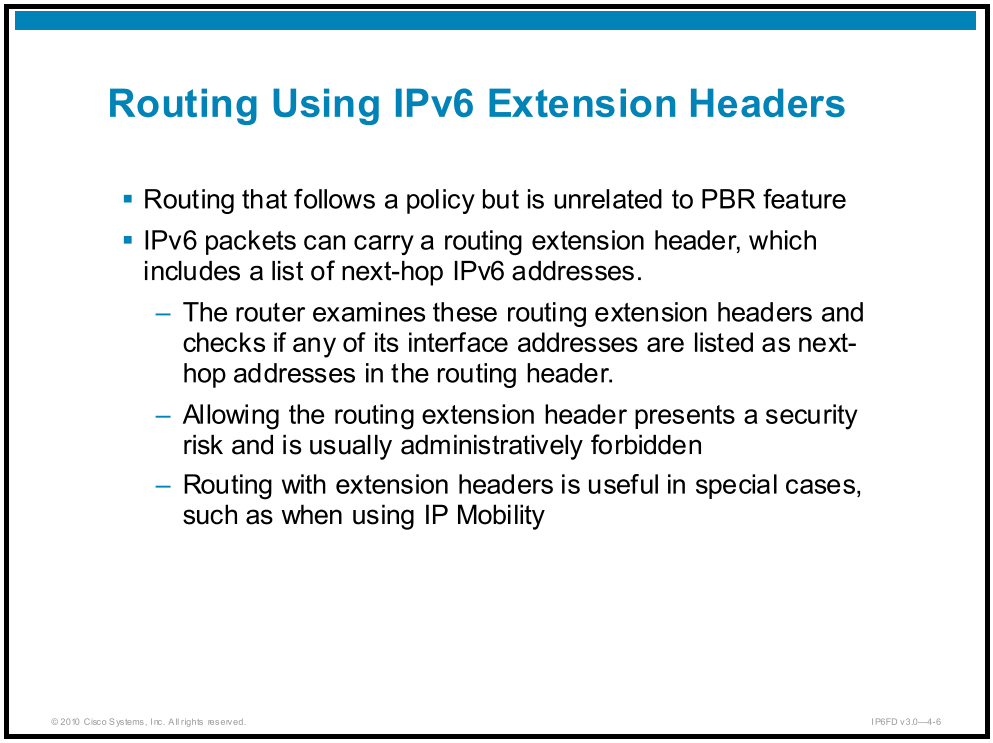
At the same time, you can use the route map to set IP precedence on a packet, which facilitates QoS operations further in the network. The packets that go to "interesting" destinations are marked with IP precedence (a 3-bit designator in the header of an IPv6 packet), based on which you can perform queuing on the outgoing interface and on other routers in the data path.

You might need to route packets through the network using PBR to allow a specific QoS through the network.

Note IPv6 QoS is not essentially different from IPv4 QoS. You can find more information about QoS mechanisms, queuing, marking, markers, and so on, in the *Implementing Cisco Quality of Service* (QOS) course.

Routing Using IPv6 Extension Headers

IPv6 offers a capability to route using the routing header, which is an extension header that is appended directly after the initial IPv6 packet header.



Routing IPv6 using routing extension headers does not follow the usual routing rules where the router, upon receiving the packet, chooses the outgoing interface that is based on the destination IP address. A routing header is an extension header that is identified by the value 43 in the IPv6 Next Header field. There are two types of the routing header:

- Type 0, which has similar functionality as that in IPv4 source routing (multiple intermediate routers)
- Type 2, which is used for Mobile IPv6 and includes the home address of a mobile node and should be allowed by the firewalls to support communication with mobile nodes

IPv6 routing extension headers allow you to set an explicit path using next-hop IPv6 addresses. When the router receives such a packet, it checks the address list in the option header and determines if one of the router interfaces has the same IP address as listed in the header. It then routes the packet through that interface.

While this can be helpful, such as when implementing IPv6 Mobility, in usual environments it is considered a security issue because you cannot implement a consistent routing policy and rules. Therefore, when routers receive these packets, they are configured to forward them using usual forwarding rules. This applies to the “Type 0” routing option header. Type 0 is deprecated as per RFC 5095 and can be completely disabled on the routers by configuring the **no ipv6 source-route** command.

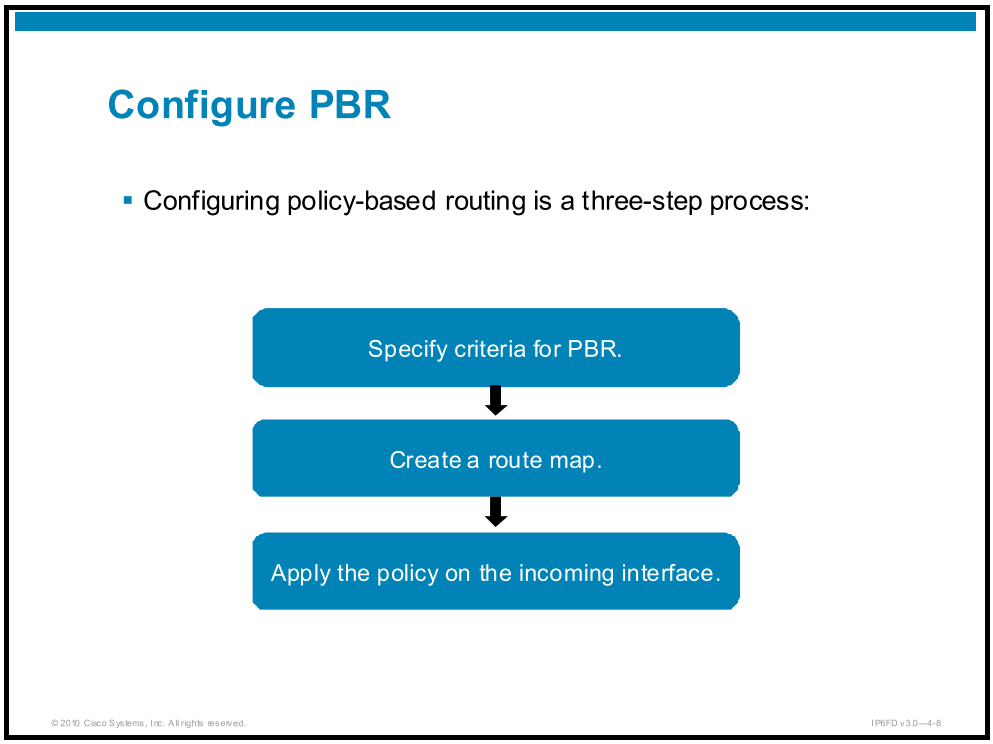
When implementing IPv6 Mobility, when a station with an IPv6 address roams into another network, the mobile station can use the optional routing header for the return traffic to go into that subnet where the mobile station currently resides. The routing option header that is used here is Type 2, and these headers are usually permitted on routers.

Note In service provider networks, explicit paths through the network can be defined using MPLS traffic engineering.

IPv6 Mobility and IPv6 option headers have dedicated sections in this course and are not discussed here. This lesson is about the PBR feature.

Configure PBR

This topic describes configuration steps in configuring PBR.



Configuring policy-based routing is a three-step process.

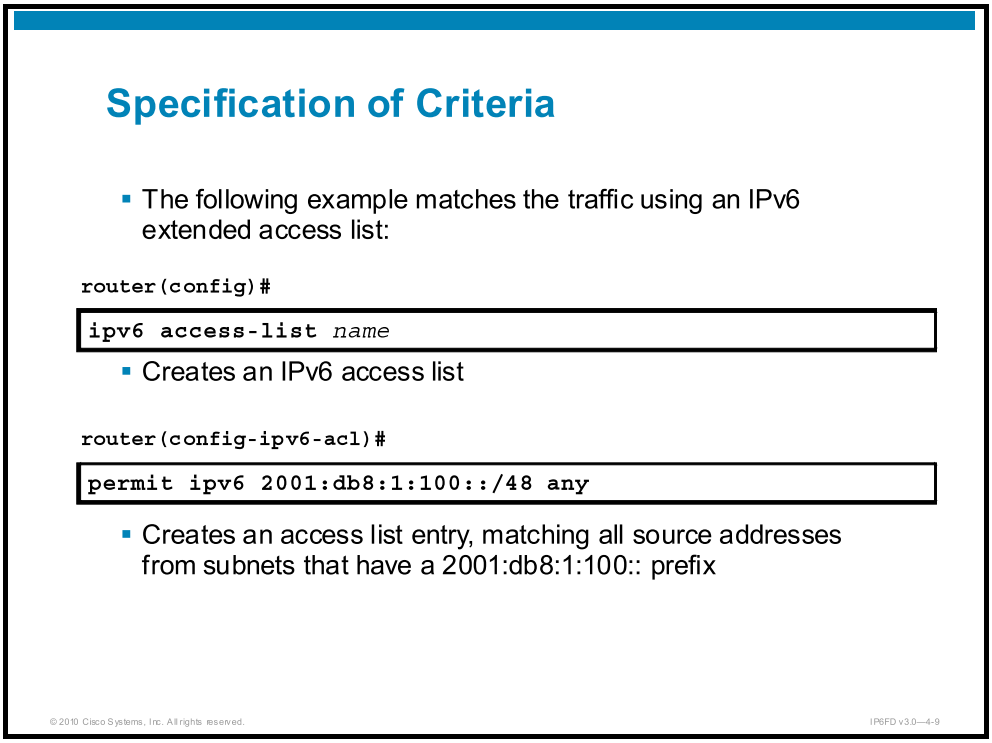
The first step is to specify the criteria for the traffic you would like to route using a special policy, that is, any other parameter than the destination IPv6 address.

The second step is to create a route map, where you combine the conditions that are specified in the first step to an action, such as how and where to route the packet.

The third step is to apply a policy where you expect to receive the traffic and route it using a PBR policy.

Specification of Criteria for PBR

This subtopic outlines how to specify the criteria for traffic that will be policy-routed using an access list.



The easiest way to specify the traffic to be matched in a route map is by using an access list. Extended access lists have the ability to match on source and destination IPv6 addresses, source, and destination ports (on the transport layer), and so on. This way, you can route selectively based on the application type as well, that is, you use some links for file transfers only or for VoIP.

In this example, all packets that are sourced from the 2001:db8:1:100::/48 networks will be matched by the access list.

PBR can classify traffic that is based on extended access list criteria. Access lists, then, establish the match criteria.

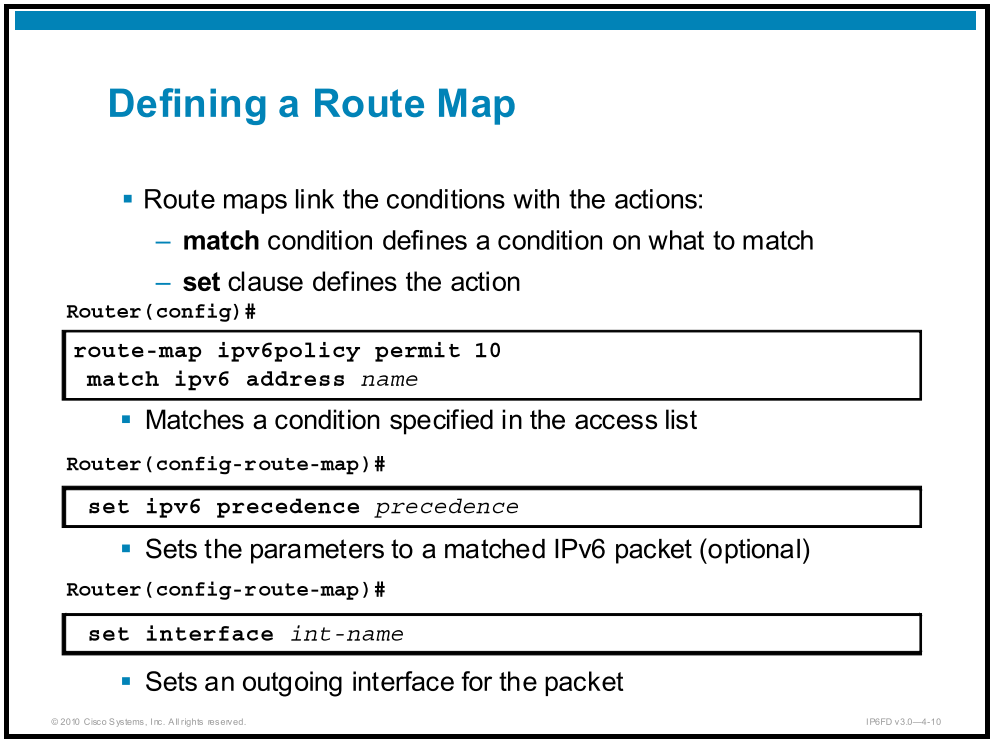
This example shows matching on source addresses; you can use PBR if you match on destination address or protocol as well, or even match on packet length using the match length statement in the route map.

In addition to matching the packets using source addresses, the packets can be matched using access lists to match on the following:

- Input interface
- Source IPv6 address (using a prefix list or a standard or extended access control 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)

Route Maps in PBR

Route maps combine the conditions for the traffic with the actions that are performed on that traffic for PBR.



As already mentioned, route maps are universal tools for filtering routes. Route maps are used to define PBR rules as well, but not all features of route maps are applicable in this case.

You can match the traffic using the following conditions regarding IPv6 addresses (using the **address** keyword).

The example shows matching on an IPv6 address, the *name* being the name of an IPv6 access list.

The route map can set IP precedence for IPv6 packets using the **precedence** keyword, so that the packets can be classified based on the IP precedence marker in the IPv6 header later on the router or further in the network.

The most commonly used option is to set a different outgoing router interface for packets that are matched by the access list. This is accomplished by the **set interface** command.

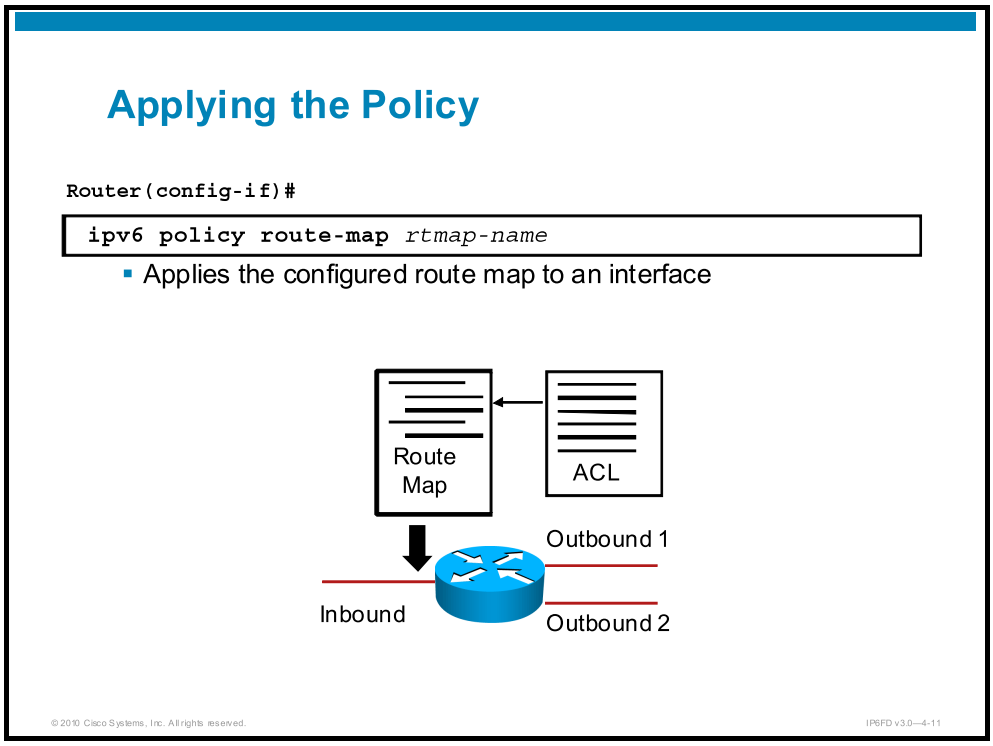
You may set multiple forwarding statements in the PBR for IPv6 route map. The following set statements may be specified:

- **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 router 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 of a specified interface. This set statement is used only when there is no explicit entry for the packet destination in the IPv6 RIB.

Applying the Policy

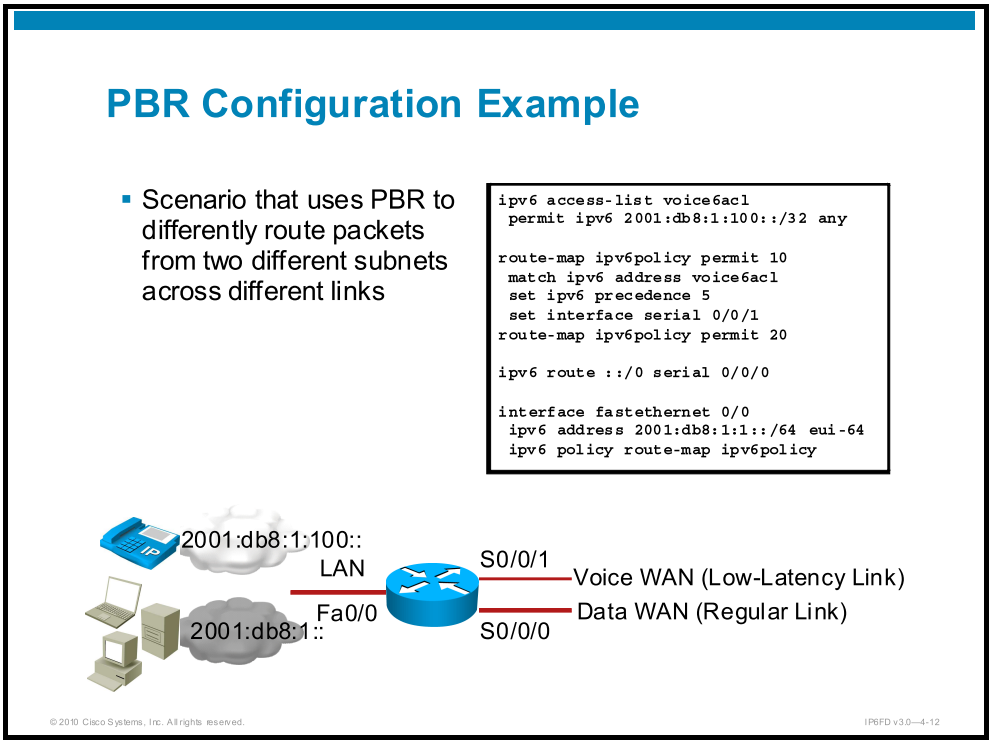
Applying the policy to the interface is the final stage of configuring PBR.



To apply a route map to an interface, you use the **ipv6 policy** command. This command is applied on the *inbound* interface where you expect the traffic. This way, you can define the *outbound* interface for the traffic.

Configuration Example

This subtopic illustrates a configuration example for a simple PBR scenario.



The example scenario matches the traffic on the router LAN interface. The traffic is mixed, coming from two source subnets: voice traffic comes from the subnet 2001:db8:1:100::, and data traffic comes from the subnet 2001:db8:1::. The router receives all packets on the FastEthernet 0/0 interface and has a default route to Serial 0/0/0 interface. However, when using PBR, you can configure routing of packets that are sourced in the 2001:db8:1:100:: subnet to be sent out on the Serial 0/0/1 interface.

Note The IPv6 addressing is simplified for illustration purposes only.

When you are using the **set interface** command in the route map for outgoing traffic, the interface must be the type “point to point.”

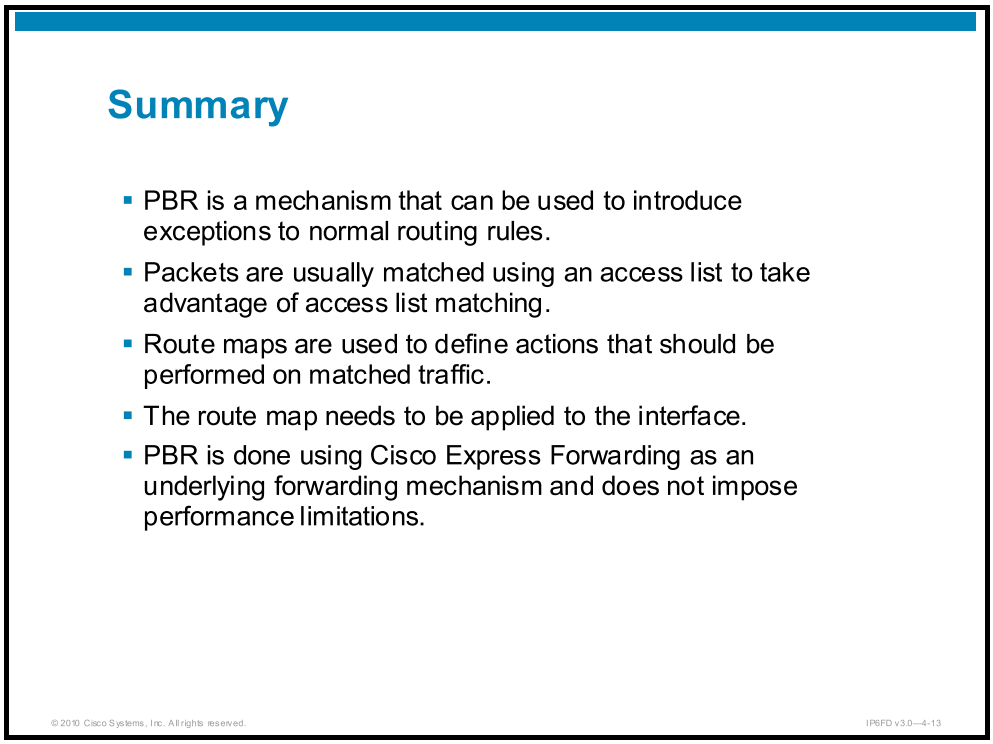
PBR and Cisco Express Forwarding

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.

Summary

This topic summarizes the key points that were discussed in this lesson.



References

For additional information, refer to this resource:

- "Implementing Policy-Based Routing for IPv6" at http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-pol-bsd_rtng_ps6441_TSD_Products_Configuration_Guide_Chapter.html#wp1055346

Lesson 7

Configuring FHRP for IPv6

Overview

This lesson describes the characteristics of First Hop Redundancy Protocol (FHRP) for Internet Protocol version 6 (IPv6). FHRPs are used to offer redundant connections on the network layer for upstream connectivity. There are two FHRPs that are used in IPv6, Hot Standby Router Protocol (HSRP) and Gateway Load Balancing Protocol (GLBP).

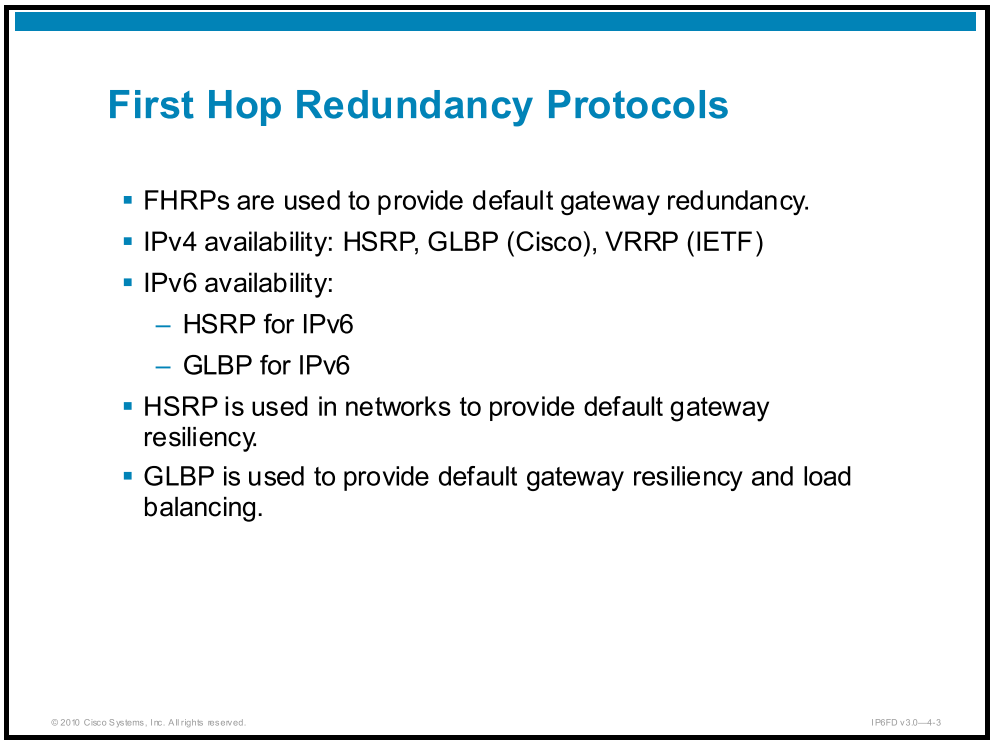
Objectives

Upon completing this lesson, you will be able to outline the concepts of FHRPs and describe HSRP and GLBP for IPv6. This ability includes being able to meet these objectives:

- Describe the concepts of FHRPs
- Describe HSRP and explain the configuration steps in configuring HSRP for IPv6
- Describe GLBP and explain the configuration steps in configuring GLBP for IPv6

First Hop Redundancy Protocols and Concepts

This topic describes the concepts of FHRPs.



FHRPs are used to provide uninterrupted communication to networks by implementing gateway redundancy. FHRPs are mainly used in access networks for clients and servers and at the edge of networks for edge router redundancy.

FHRPs include the HSRP, GLBP, and Virtual Router Redundancy Protocol (VRPP). The first two are proprietary to Cisco, while the latter is standardized on the Internet Engineering Task Force (IETF). All of these protocols are available for IPv4. Currently, only HSRP and GLBP are available for IPv6.

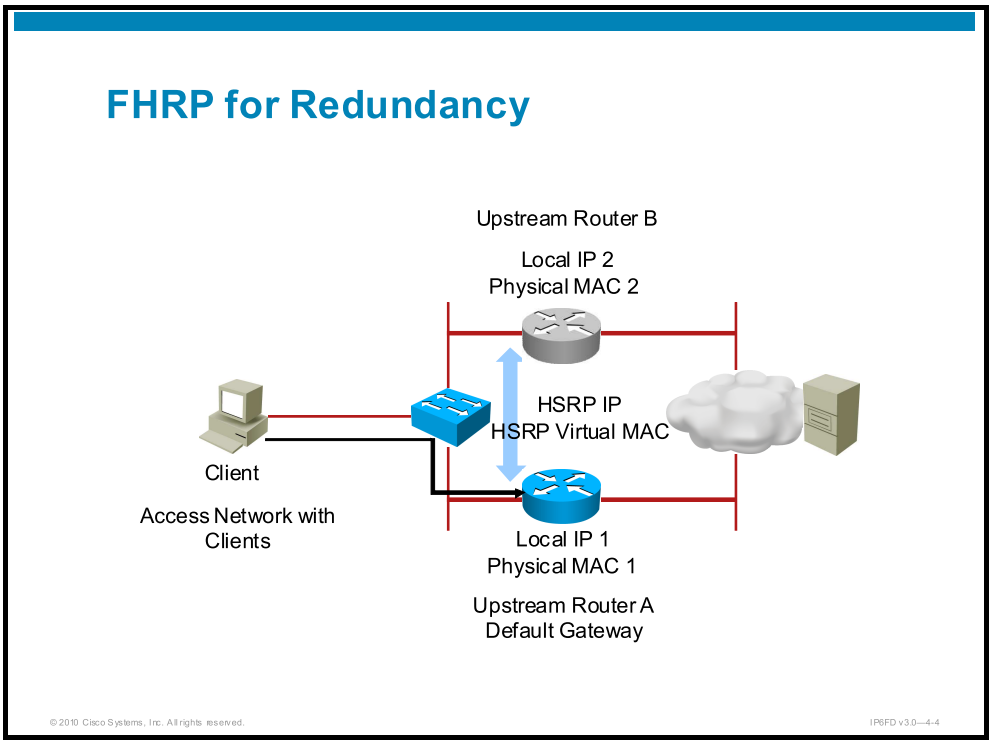
HSRP requires at least two routers to work in an HSRP group, where one router acts as an active gateway and the other router is in standby, taking over if the primary router fails. The primary router only forwards data.

GLBP offers gateway load balancing, where more than one router forwards the packets upstream on the network. This way, routers accomplish load balancing of available uplinks; return traffic normally flows only through a single router.

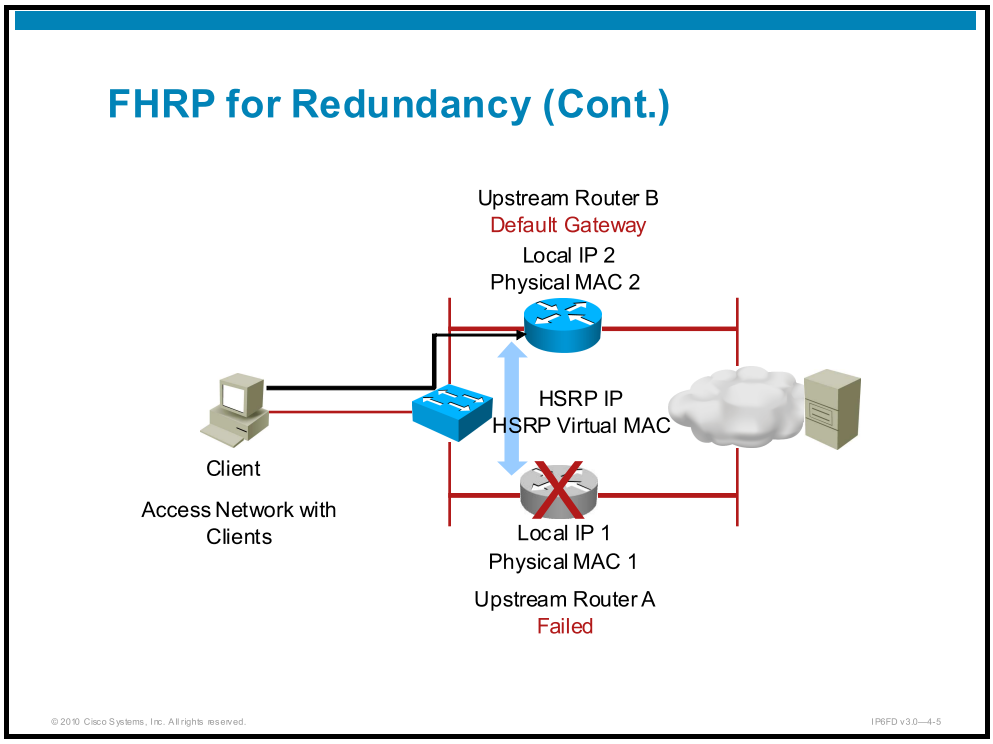
Note In some cases, HSRP can be configured so that the standby router forwards the traffic upstream as well. This is in the example of the Nexus 7000 data center switch running NX-OS software.

FHRP for Redundancy

FHRPs can be used to provide redundancy for access networks.



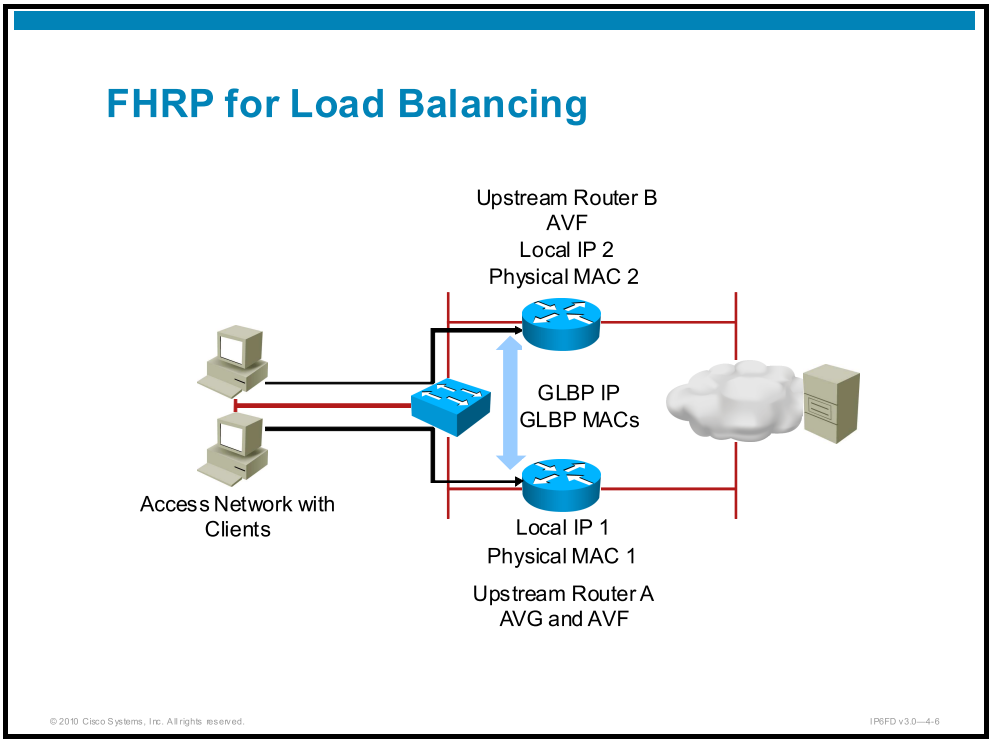
The figure illustrates the normal operation of FHRP. In HSRP, one router is active and “holds” the virtual IP address. This applies to both IPv4 and IPv6. HSRP functions use a virtual IP and a virtual MAC address, which the clients refer to when sending data to the network. Both routers exchange HSRP hello packets to maintain the HSRP group in the “up” state.



This figure illustrates the failure of the HSRP primary router. The secondary router will take over the packet forwarding, moving the virtual IP and virtual MAC addresses to itself.

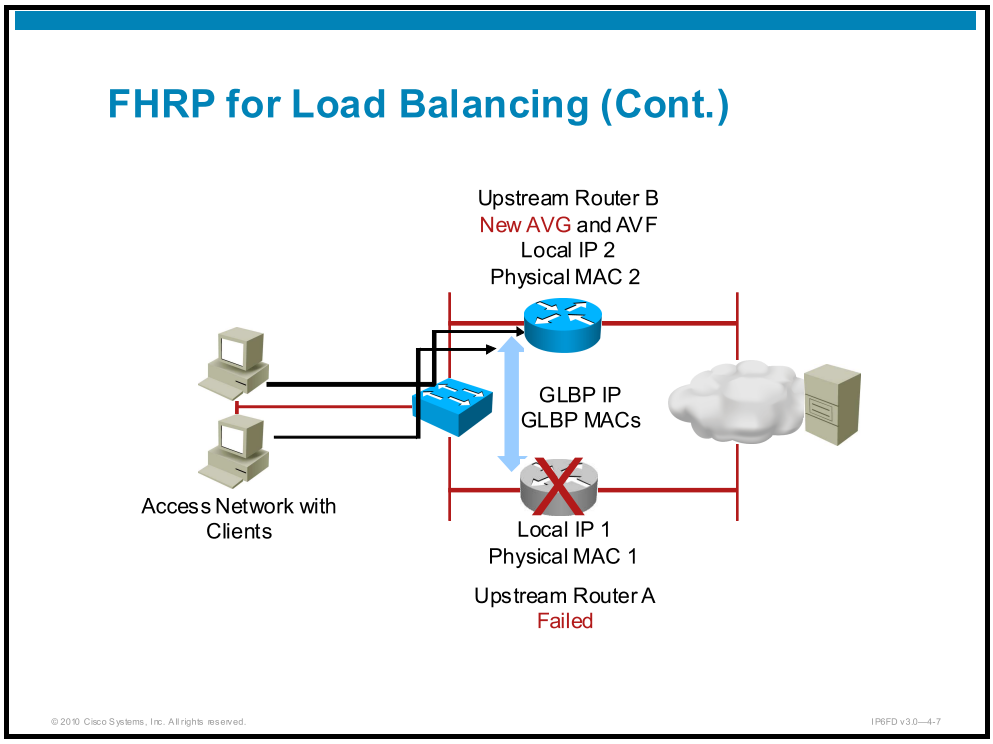
FHRP for Load Balancing

Another use of FHRP is to provide redundancy and load balancing for large access networks.



Default gateway load balancing becomes applicable in large access networks. IPv6 is able to accommodate a larger number of devices in networks, with the lower 64 bits of the IPv6 address that is reserved for the host portion. This means that the access subnets can potentially be larger and contain more devices, which eventually generate much more traffic. Using GLBP, you can configure several routers to act as default gateways for these networks and share the load when forwarding the traffic upstream.

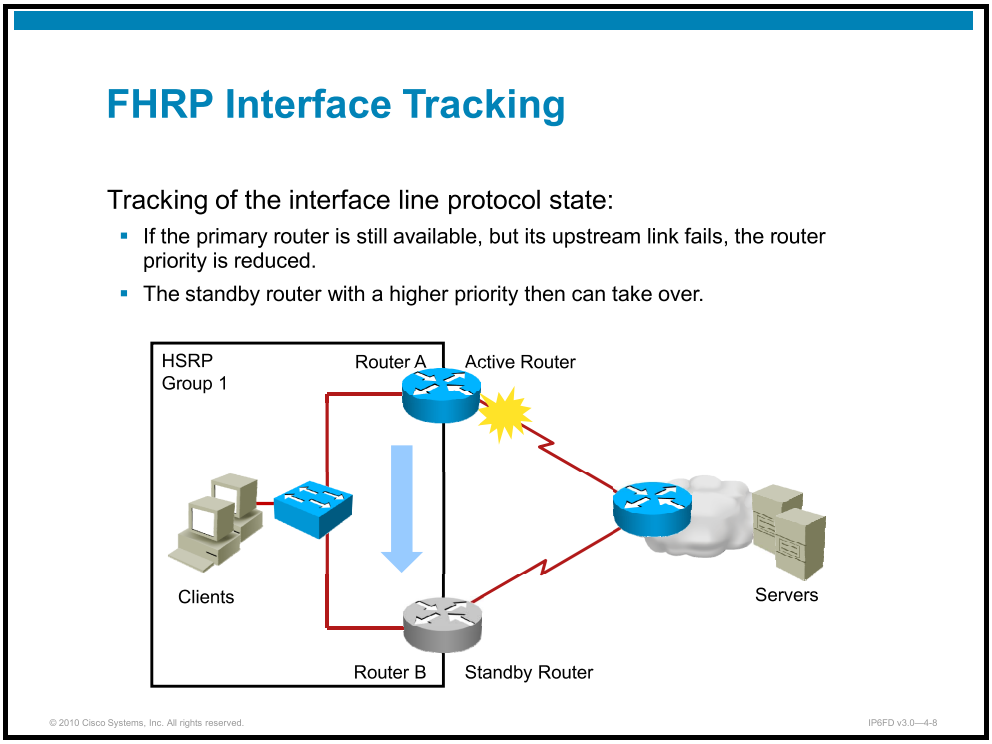
All routers forward traffic, and they are called active virtual forwarders (AVFs) in GLBP terminology. Only one router is responsible for load distribution, and it is called the active virtual gateway (AVG). All clients are configured with the IP address of the AVG as the default gateway address.



If there is an AVG failure, another router becomes the AVG. If an AVF fails, the load on that AVF router is distributed to other AVF routers. Every AVG router is an AVF router as well.

Interface Tracking

This subtopic explains the concept of interface tracking.

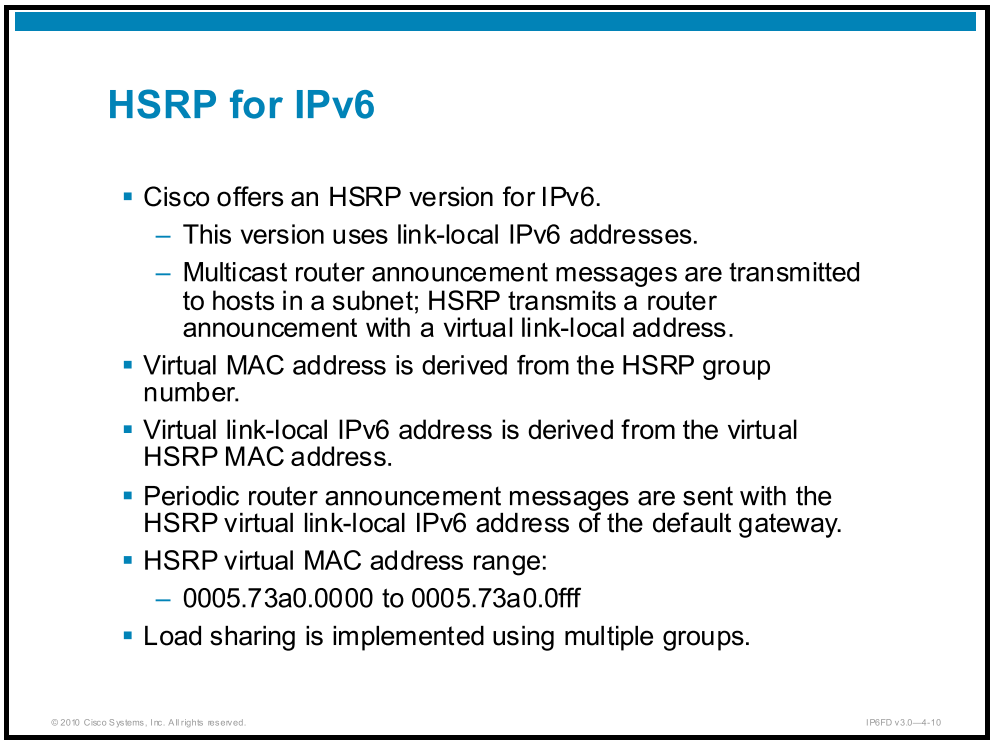


The interface tracking feature of FHRPs is widely used, as it is often more probable that the router uplink would fail rather the router itself. However, without using interface tracking, FHRP would be unable to detect the loss of a link.

The “hello messages” that the routers send to each other to maintain the FHRP group serve only to detect a failed router but not a failed link. When using interface tracking to manipulate router priority, the router with highest priority can naturally take control of the group and be in the forwarding path. Thus, FHRP role pre-emption must be enabled on all routers in the group.

HSRP for IPv6

This topic describes HSRP and the configuration steps in configuring HSRP for IPv6.



Cisco offers two FHRPs for the IPv6 protocol, HSRP and GLBP. HSRP and GLBP function similarly to their IPv4 counterparts, with a few specifics based on IPv6.

HSRP for IPv6 uses link-local IPv6 addresses instead of global addresses.

IPv6 hosts learn of available IPv6 routers through IPv6 neighbor discovery Router Advertisement (RA) messages. These messages are multicast periodically, or they may be solicited by hosts. HSRP is designed to provide only a virtual first hop for IPv6 hosts. The hosts autoconfigure themselves to use a default gateway by learning the IPv6 address from the RA messages. This is in contrast to IPv4, where an IPv4 default gateway must be configured manually.

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.

RAs are sent only by the active HSRP 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.

HSRP Groups

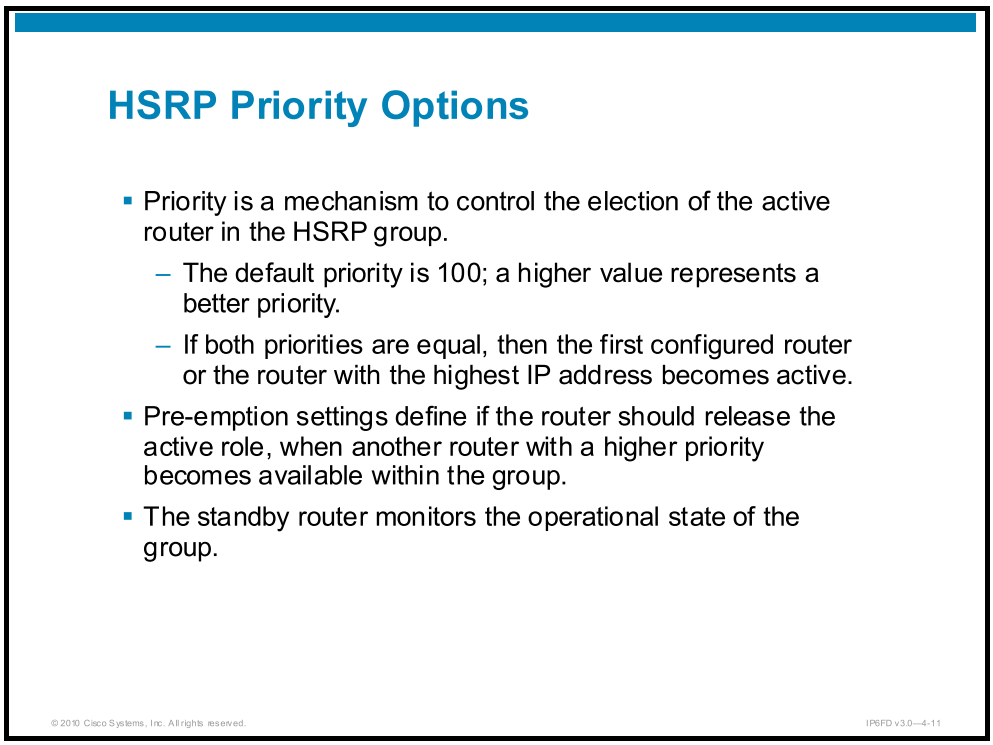
HSRP for IPv6 is configured on a group of routers that form an HSRP group. The same interface can have multiple HSRP groups that are enabled to provide HSRP load sharing (for example, for different VLANs, and so on). Based on the group number, an HSRP virtual MAC address is derived, and from that address, an HSRP virtual link-local IPv6 address is derived.

Load sharing on a LAN segment can be implemented using multiple HSRP groups, where each group holds a virtual default gateway address. Assuming there are two routers on that LAN segment, one router operates as active for the first HSRP group and standby for the second group, while the other router operates as standby for the first group and active for the second HSRP group.

Upon failure of one of the routers, the remaining router takes the load for both groups. As a downside, approximately half of the clients need to have the virtual IP address of the first HSRP group that is set as the default gateway, and the other half of the clients need to have the second virtual IP address of the HSRP group that is set as the default gateway. Load distribution is manual in this case.

HSRP Priority and Object Tracking

This subtopic describes enhancements to HSRP, such as priority configuration and object tracking configuration.



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 Election Process

When an HSRP group is configured, routers first listen for hello messages. After a period of time, the routers start to compete for the role of an active router, and the election process begins. Routers agree on who will become the active router. This role is assumed by the router with the highest priority value, or, if there are equal priorities, the one with the highest IP address for the respective group.

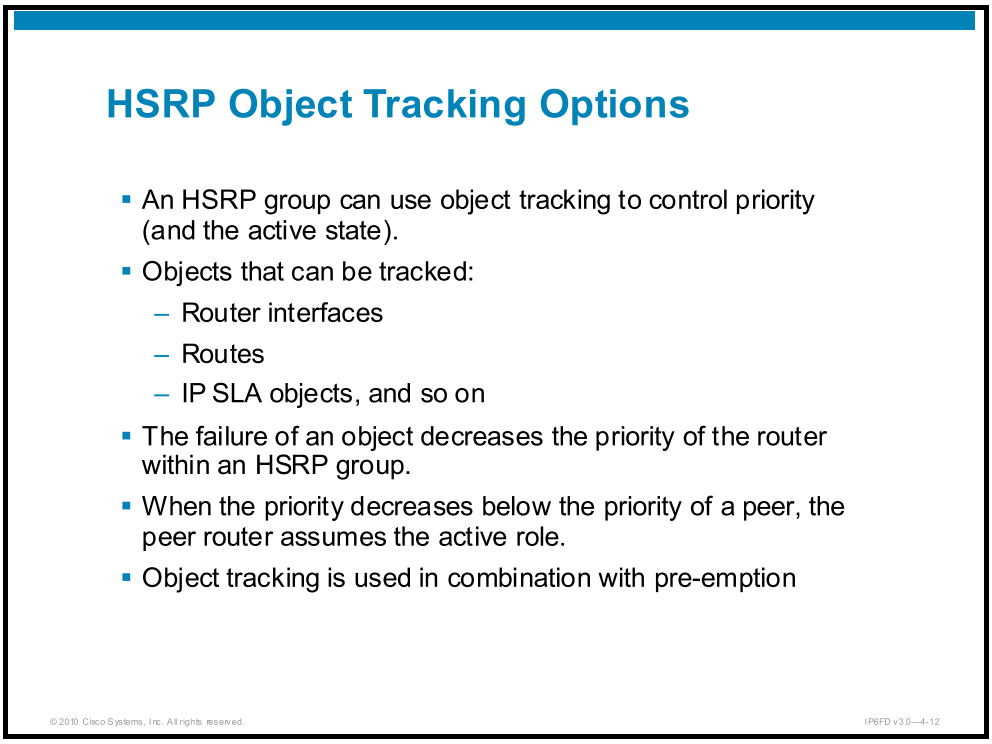
Note	The router that is configured first becomes the active router for that group. Configuring another router with a higher IP address afterward without the **preempt** option does not make that router an active router. The router that is already active remains active despite the lower IP address.

At the same time, another router is elected as the standby router (the router with the second-highest IP address becomes the standby router).

Once the election process is finished, there is one active router and one standby router in the HSRP group and other routers (if there are more than two routers in the group) that listen.

To minimize network traffic, only the active and standby routers send periodic HSRP messages.

The standby router also monitors the operational status of the HSRP group.



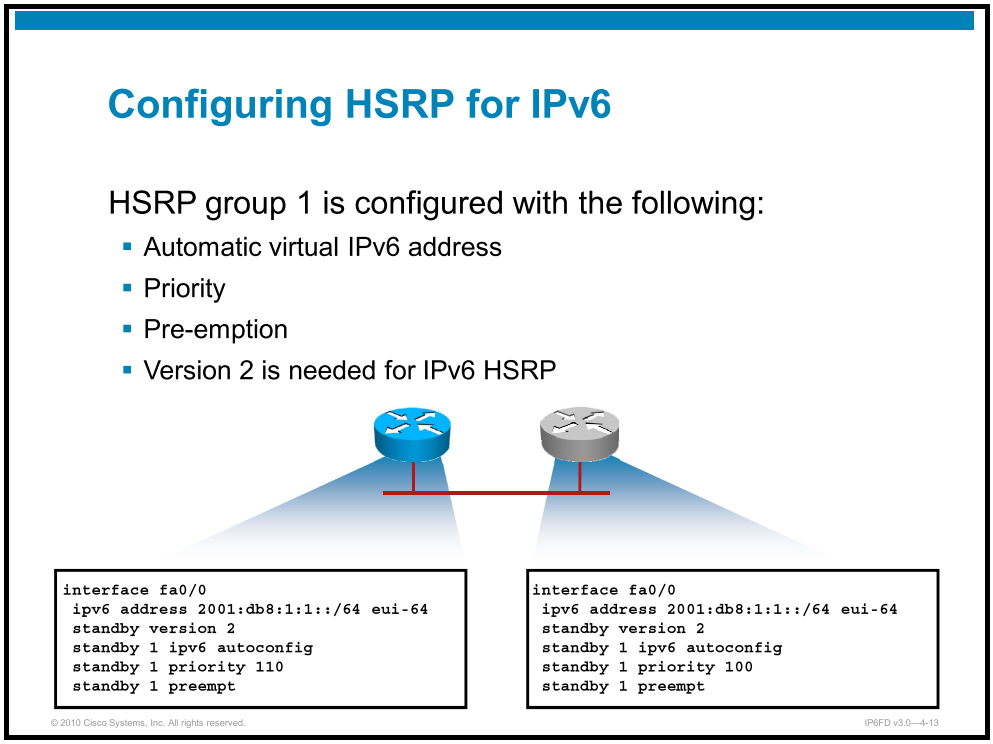
The standby tracking feature of HSRP enables customers to proactively cut over to a standby HSRP device if the upstream connection goes down (but the HSRP device is still active).

Apart from interface tracking, HSRP also supports the tracking of different objects that can influence the active role selection:

- **Interface line-protocol state:** The interface line-protocol state is the same as the HSRP tracking operation in prior releases. The tracking process is configured to track the line-protocol state of the interface.
- **Interface routing state:** A tracked IP routing object is considered operational when the platform is routing IP, the interface line-protocol is operating, and IP routing is enabled and active on the interface.
- **State of an IP route (reachability):** A tracked IP route object is considered operational and reachable when a routing table entry exists for the route and the route is not inaccessible.
- **IP route metric threshold:** The IP route metric threshold tracks the scaled metric value of an IP route to determine if it is above or below a threshold.
- **IP service level agreement (SLA) operations:** HSRP tracks IP SLA operations for reachability and thresholds.

Configuring and Monitoring HSRP for IPv6

This subtopic describes how to configure and monitor HSRP for IPv6.



This figure shows a sample configuration of an HSRP group with two routers that share the same LAN segment.

For IPv6 functionality, HSRP must be set to operate the HSRP version 2.

The easiest way is to use the **autoconfig** keyword. The router will generate a virtual MAC address from the HSRP group number (the number after the **standby** keyword), and the router will derive a link-local IPv6 address from it.

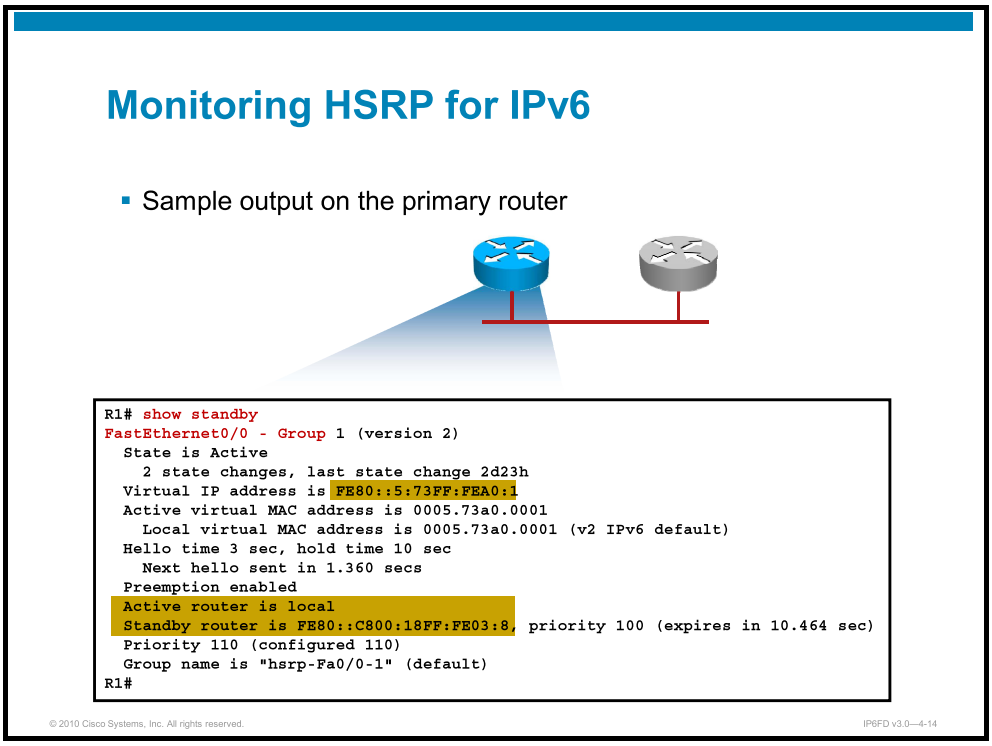
Priority and pre-emption settings are set so that the left router has a higher priority than the right one, and the left router will assume the active role whenever it is present (the **priority** and **preempt** keywords).

Note HSRP for IPv6 can be configured using global addresses as well. This is useful for injecting the HSRP default gateway address in a routing protocol where it is being carried for several hops. If the address is link-local, this route is useless outside the local Layer 2 scope.

For access subnets (or LAN segments), HSRP is used with link-local addresses.

Monitoring HSRP

This subtopic describes how to monitor HSRP for IPv6.



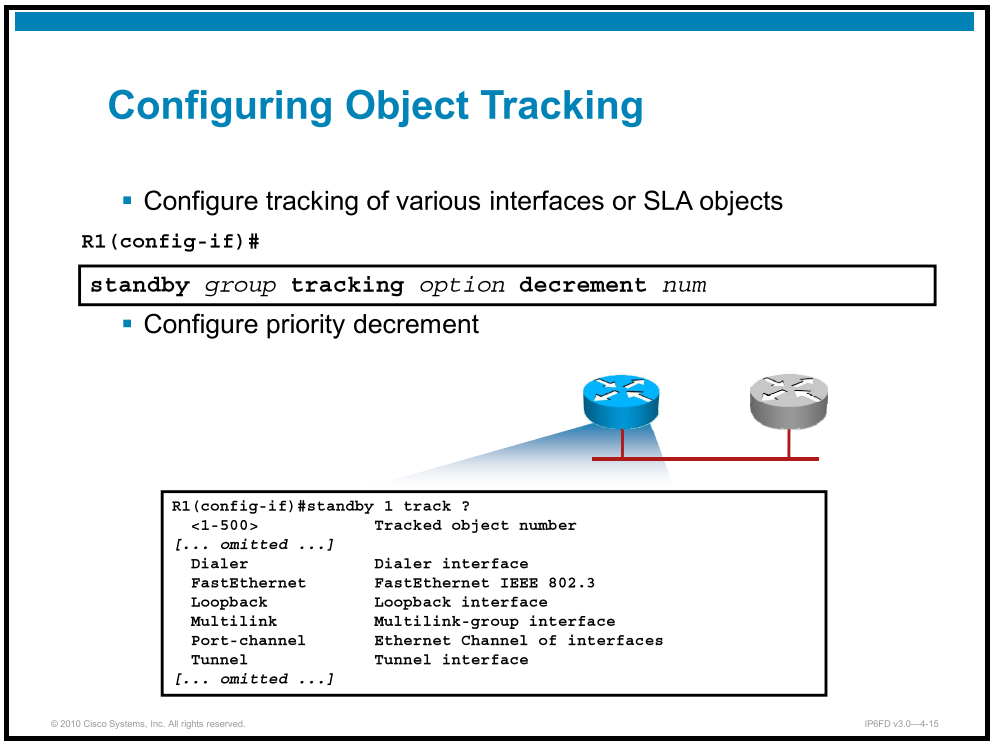
The **show standby** command is used to monitor the state of an HSRP group. The example is taken from an active HSRP router.

The active HSRP router also sends IPv6 route advertisement messages, showing the virtual IPv6 address as the default gateway:

```
*Apr 15 14:05:00.149: ICMPv6-ND: Sending RA from FE80::5:73FF:FEA0:1 to
FF02::1 on FastEthernet0/0
*Apr 15 14:05:00.153: ICMPv6-ND:      MTU = 1500
*Apr 15 14:05:00.157: ICMPv6-ND:      prefix = 2002:1::/64 onlink autoconfig
*Apr 15 14:05:00.157: ICMPv6-ND:               2592000/604800 (valid/preferred)
```

Configuring Object Tracking

This subtopic explains how to configure HSRP object tracking.



HSRP object tracking is configured using the **standby track** command along with a configured **decrement** value. Tracking is used to invoke a switchover even if the primary router is still available on the LAN, but the primary router had lost its uplinks and therefore it does not make sense to use it to carry traffic upstream.

You can track interfaces (configured directly) or track objects that are configured by a tracking object first. If you wish to track an IP SLA probe, such as a probe that constantly pings an IP address, you have to bind the probe to a tracking object first. The probe and tracking objects are configured indirectly.

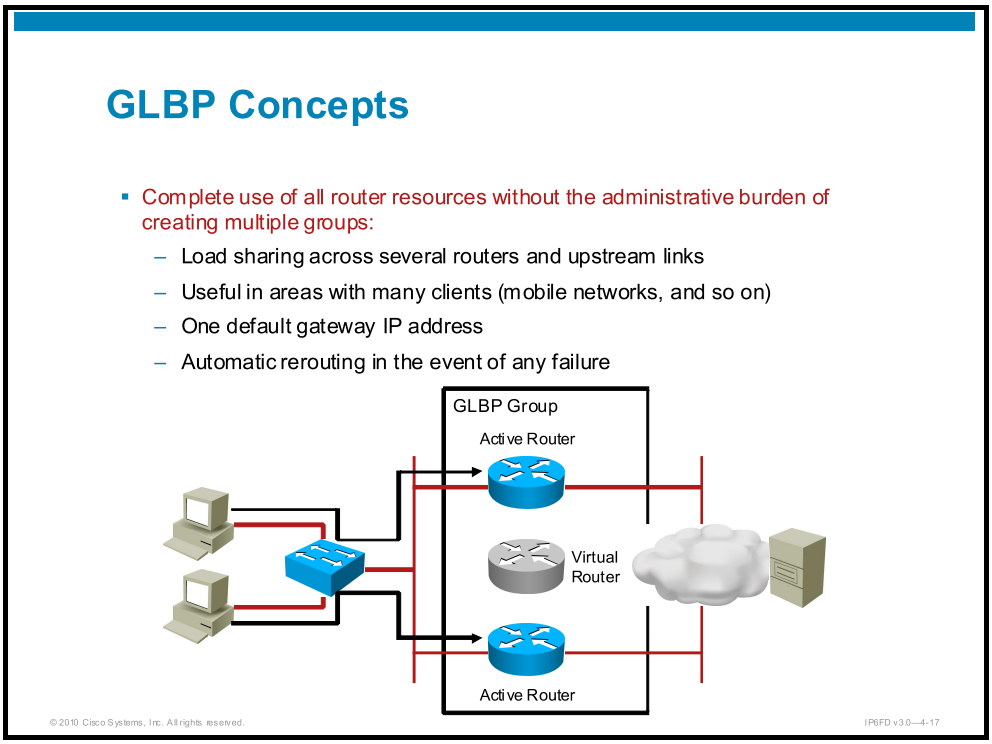
The decrement value specifies how much the priority of the primary router should be reduced to switch over to the secondary router. The decrement value is subtracted from the HSRP priority value, and when this value is below the priority of the neighbor, the neighbor assumes the active role.

Most commonly, member interfaces of an EtherChannel or physical interfaces are tracked.

Note To learn more about HSRP object tracking, refer to the IPv4 HSRP configuration guides.

GLBP for IPv6

This topic describes GLBP and the configuration steps in configuring GLBP for IPv6.



GLBP is a protocol that is developed by Cisco that is used to overcome the limitations of HSRP and Virtual Router Redundancy Protocol (VRRP) by adding load-sharing functionality.

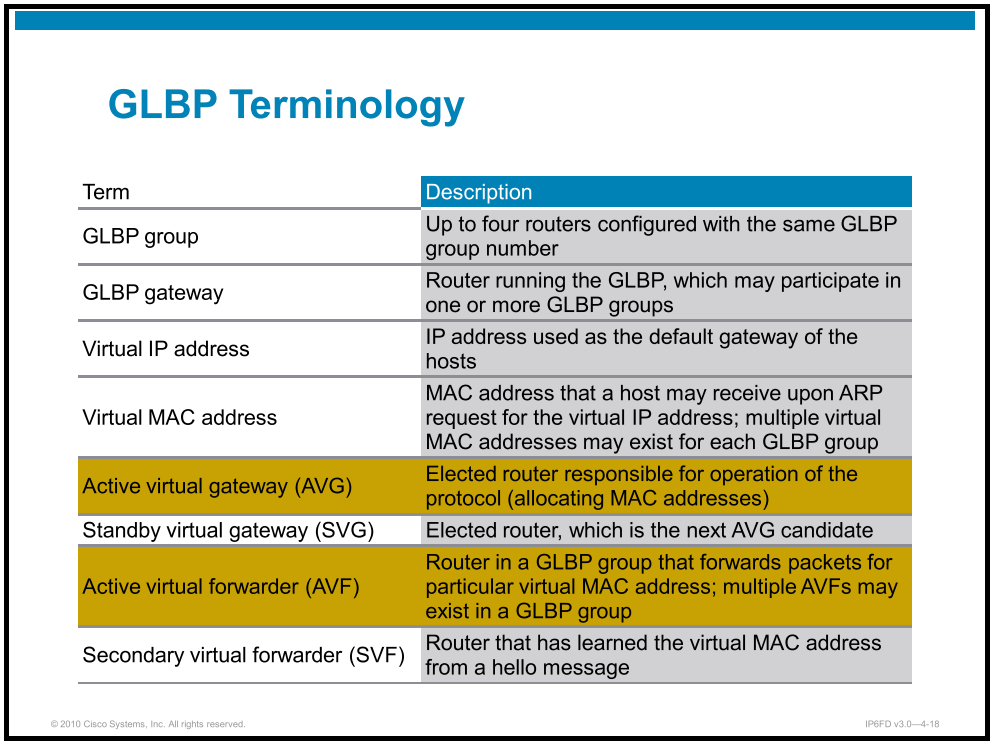
GLBP performs a similar function for the user as HSRP. HSRP allows multiple routers to participate in a virtual router group that is configured with a virtual IPv6 address.

HSRP standby routers have bandwidth that is not used. Multiple virtual router groups can be configured for the same set of routers; the hosts must be configured for different default gateways (pointing to one router or another), which results in an extra administrative burden.

The advantage of GLBP is that it 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 managed by a single router, while the other routers stand idle.

GLBP Terminology

The following terms are used with GLBP.



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 (SVF).

Active Virtual Gateway

Members of a GLBP group elect one gateway to be the AVG for that group. Other group members provide backup for the AVG when the AVG becomes unavailable. The AVG assigns a virtual MAC address to each member of the GLBP group. Each gateway assumes responsibility for forwarding packets that are sent to the virtual MAC address, which is assigned to the gateway by the AVG. These gateways are known as AVFs for their virtual MAC address.

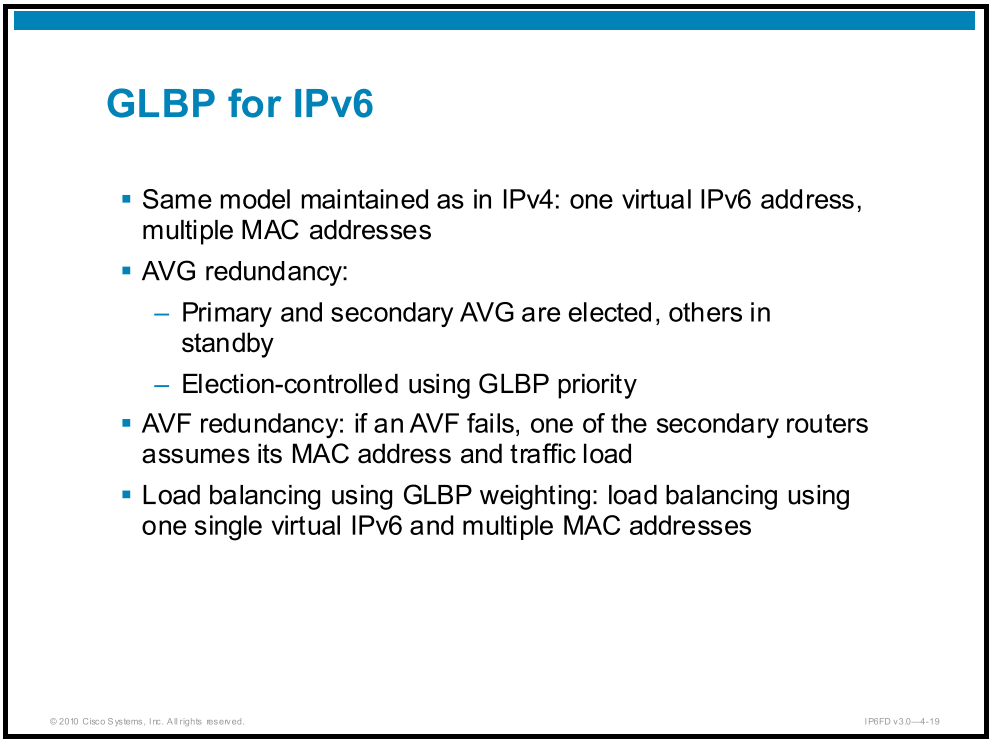
The AVG is responsible for answering Address Resolution Protocol (ARP) requests for the virtual IP address. The AVG achieves load sharing by replying to the ARP requests with different virtual MAC addresses.

Active Virtual Forwarder

An individual AVF is assigned for each virtual MAC address to forward the traffic for that virtual MAC address.

GLBP for IPv6

This subtopic describes GLBP specifics for IPv6.



The GLBP 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.

Each host is configured with the same virtual IPv6 address, and all routers in the virtual router group participate in forwarding packets.

AVG and AVF

GLBP assigns the primary router as the AVG and the secondary router as the AVF. There can be multiple AVFs that are active at the same time but only one AVG.

The benefits of using GLBP are as follows:

- **Fast failover:** GLBP detects failures fast and reacts quickly so that end devices and applications can continue to run as if no failure occurred.
- **Simple configuration:** GLBP was designed to be easy to configure and to be very similar to HSRP.

GLBP fully utilizes resources (available bandwidth) without administrative burden by load balancing across several routers while configuring only one default gateway on the servers.

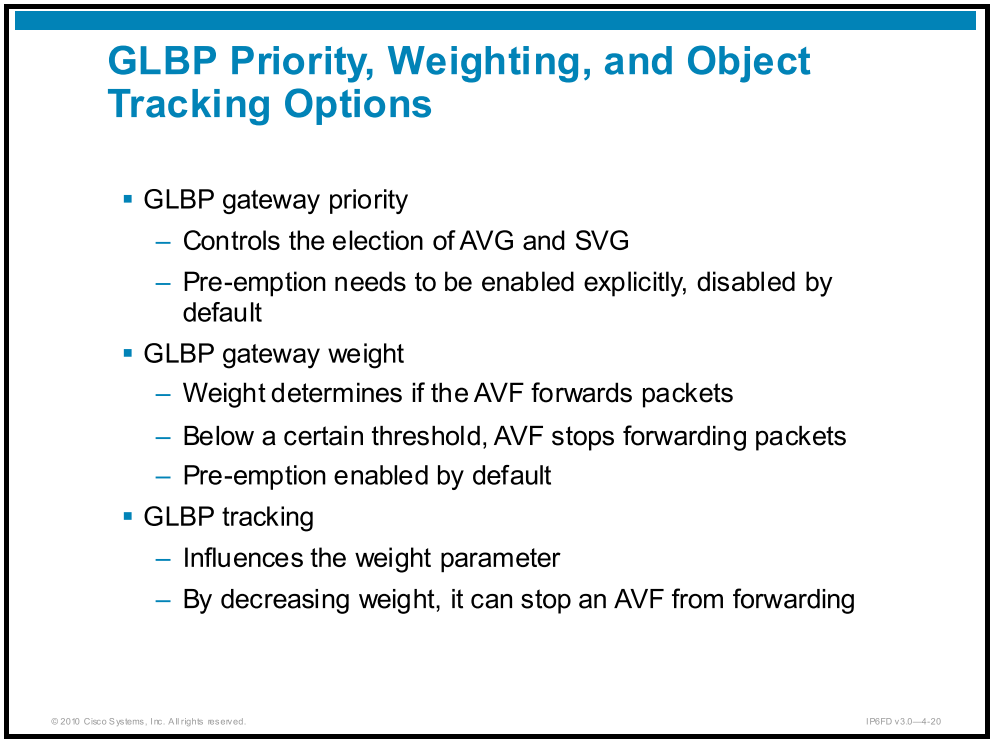
In addition to being able to set priorities on different gateway routers, GLBP also allows a weighting parameter to be set. Load balancing is not based on traffic load, but rather on the number of hosts that will use each gateway router.

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.

GLBP Priority and Object Tracking

This subtopic explains the relationship among GLBP priority, weight, and tracking.



GLBP Gateway Priority

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 (SVG), and the remaining gateways are placed in a listen state.

The AVG is responsible for assigning virtual MAC addresses to the default gateway IPv6 address.

By default, the GLBP virtual gateway pre-emptive scheme is disabled. A backup virtual gateway can become the AVG only if the current AVG fails, regardless of the priorities that are assigned to the virtual gateways. You can enable the GLBP virtual gateway pre-emptive scheme using the **glbp preempt** command. Pre-emption 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 Weight

GLBP virtual forwarder redundancy is similar to virtual gateway redundancy with an AVF. If the AVF fails, one of the SVFs in the listen state assumes responsibility for the virtual MAC address. GLBP migrates hosts away from the old forwarder.

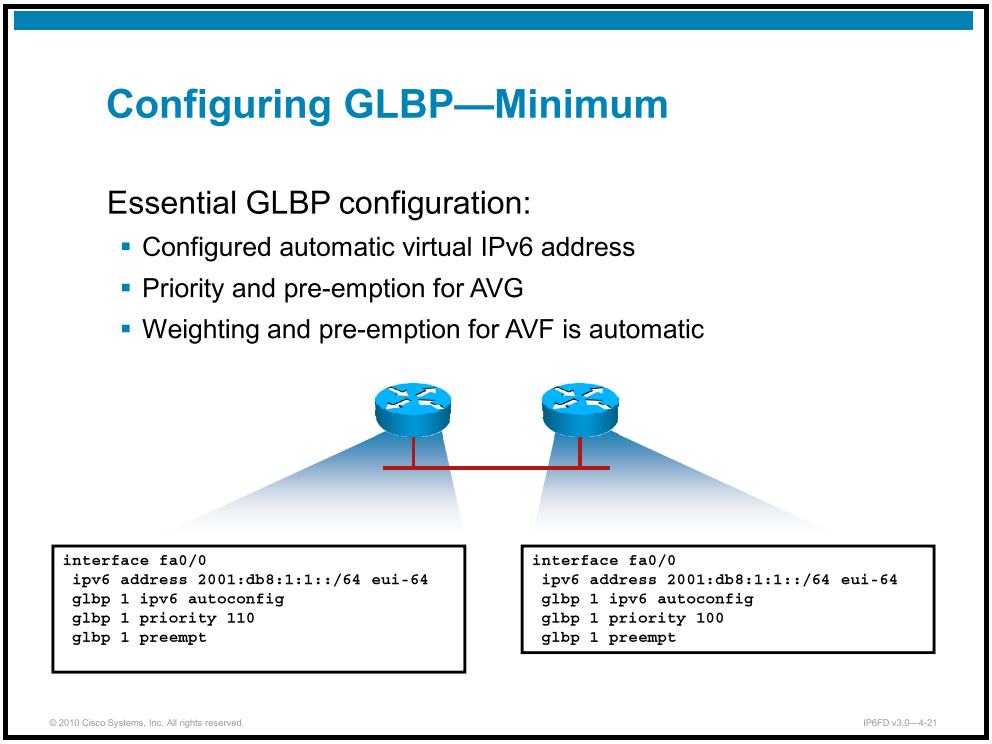
GLBP uses a weighting scheme to determine the forwarding capacity of each router in the GLBP group. The weighting that is assigned to a router in the GLBP group can be used to determine whether it will forward packets and, if so, the proportion of hosts in the LAN for which it will forward packets.

GLBP Tracking

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.

Configuring GLBP for IPv6

This subtopic explains how to configure GLBP.



The GLBP is configured using **glbp** commands in the router interface configuration mode. Each GLBP group is identified by the group number [“1”].

The GLBP gateway priority determines the role that each GLBP gateway plays and the results if the AVG fails.

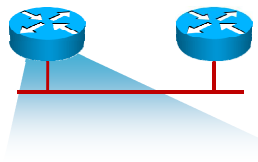
Priority also determines if a GLBP router functions as a backup virtual gateway and the order of ascendancy to becoming an AVG if the current AVG fails. You can configure the priority of each backup virtual gateway with a value of 1 through 255 using the **glbp priority** command.

The **glbp preempt** command regulates AVG (not AVF) pre-emption.

Configuring GLBP—Advanced

GLBP weighting with tracking:

- Configured automatic virtual IPv6 address
- AVF weighting configured
  - Weight below 95—cease AVF
  - Weight above 105—resume AVF role
- Object tracking configured to influence weight
  - Failure of tracking object will decrease weight by 5



```
interface fa0/0
 ipv6 address 2001:db8:1:1::/64 eui-64
 glbp 1 ipv6 autoconfig
 glbp 1 priority 110
 glbp 1 preempt
 glbp 1 weighting 110 lower 95 upper 105
 glbp 1 weighting track 2 decrement 5
 glbp 1 forwarder preempt

track 2 interface serial 1/0 line-protocol
```


By default, the GLBP virtual forwarder pre-emptive 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.

Note You can disable the GLBP forwarder pre-emptive scheme using **the no glbp forwarder preempt** command or change the delay using the **glbp forwarder preempt delay minimum** command.

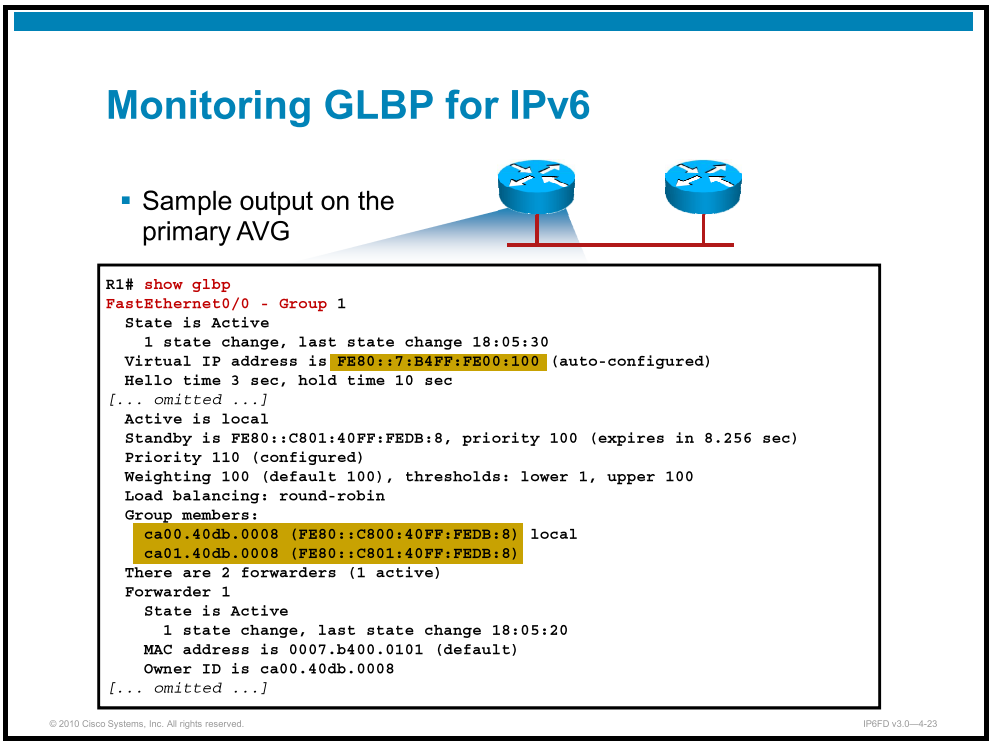
The command **track** (at the last line) configures a tracking object; in this case, it is an interface to be tracked where changes in the state of the line protocol will affect the weighting of a GLBP gateway. This command configures the interface and corresponding object number (2) to be used with the **glbp weighting track** command. The **decrement** value argument specifies a reduction in the weighting of a GLBP gateway when a tracked object fails.

The **glbp weighting** command specifies the initial weighting value, as well as the upper and lower thresholds, for a GLBP gateway. This command 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. This command is enabled by default and has a delay of 30 seconds. If the weighting value is below 95, the router will cease to be the AVF. If the weighting value rises above 115, the router will resume the AVF role and will forward traffic.

Note For more information, refer to *Configuring First Hop Redundancy Protocols in IPv6* in Cisco IOS configuration guides (listed under *Resources*).

Monitoring GLBP

This subtopic describes the monitoring commands for GLBP.



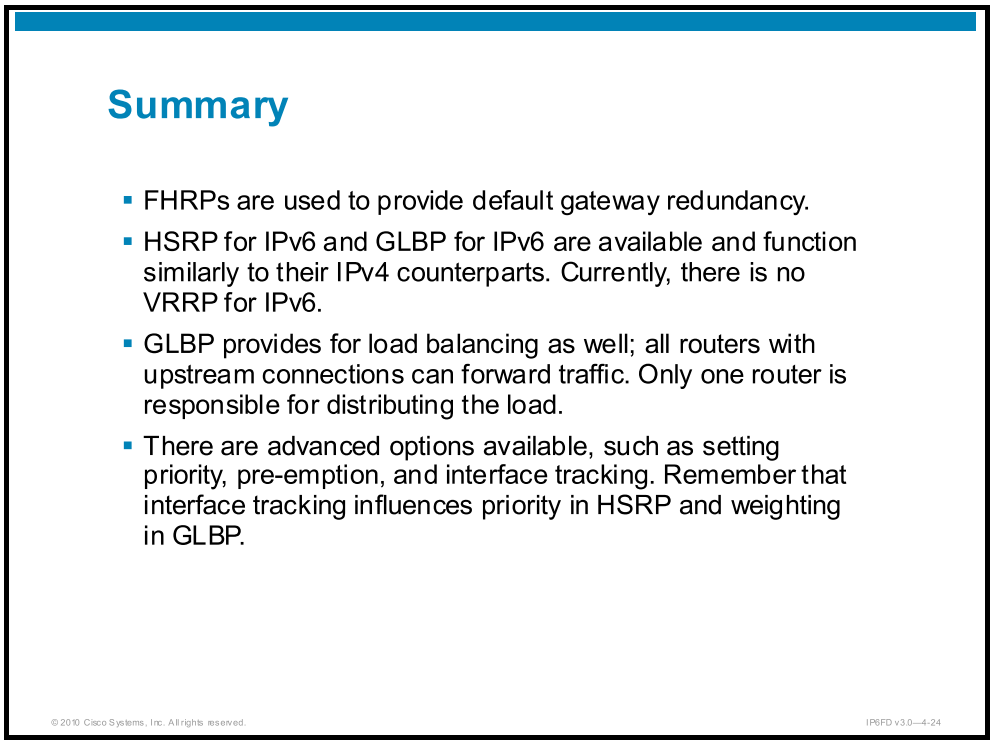
The most useful commands to monitor GLBP operation are **show glbp** and, in the case of interface tracking, **show track**.

For in-depth monitoring of GLBP, you can use the following debug commands:

- **debug condition glbp**
- **debug glbp errors**
- **debug glbp events**
- **debug glbp packets**

Summary

This topic summarizes the key points that were discussed in this lesson.



References

For additional information, refer to these resources:

- *Cisco IOS IPv6 Configuration Guide, Release 12.4T,* "Configuring First Hop Redundancy Protocols in IPv6" http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-fhrp.html#wp1055254
- *Cisco IOS IP Application Services Configuration Guide, Release 12.4,* "FHRP Features Roadmap" http://www.cisco.com/en/US/docs/ios/ipapp/configuration/guide/ipapp_fhrp_rm_ps6350_TSD_Products_Configuration_Guide_Chapter.html#wp1063089

Lesson 8

Configuring Route Redistribution

Overview

This lesson discusses redistribution of IP version 6 (IPv6) routing information, differences among various routing protocols, and changes in the behavior of redistribution compared to IP version 4 (IPv4).

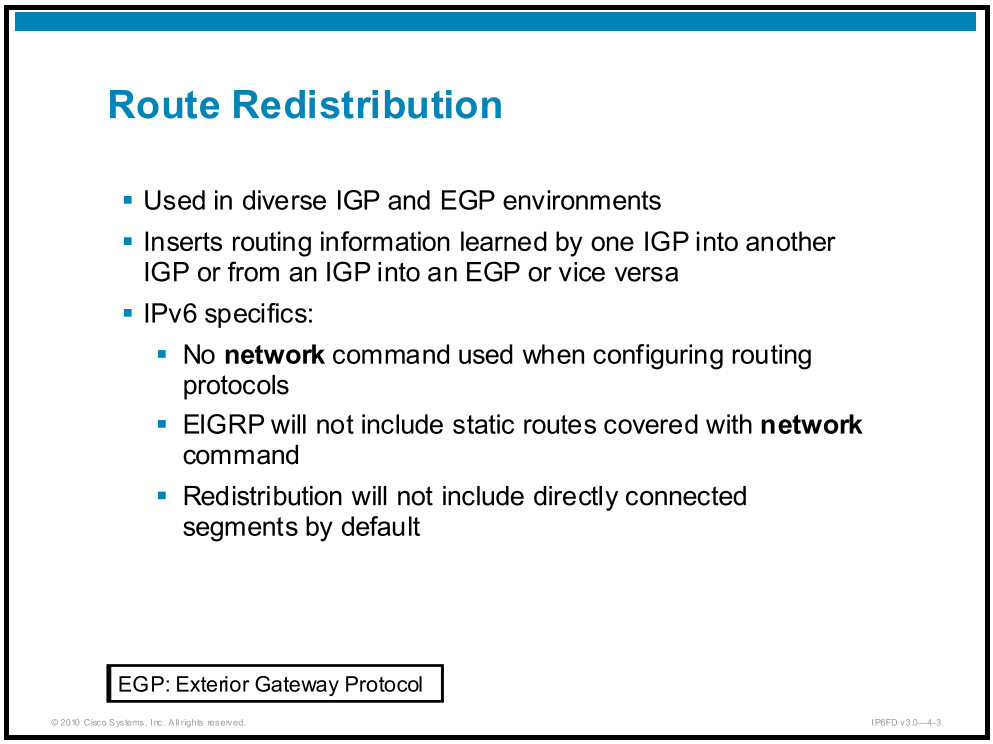
Objectives

Upon completing this lesson, you will be able to describe route redistribution. This ability includes being able to meet these objectives:

- Describe route redistribution
- Describe PE-CE redistribution for service providers

Route Redistribution

This topic describes route redistribution.



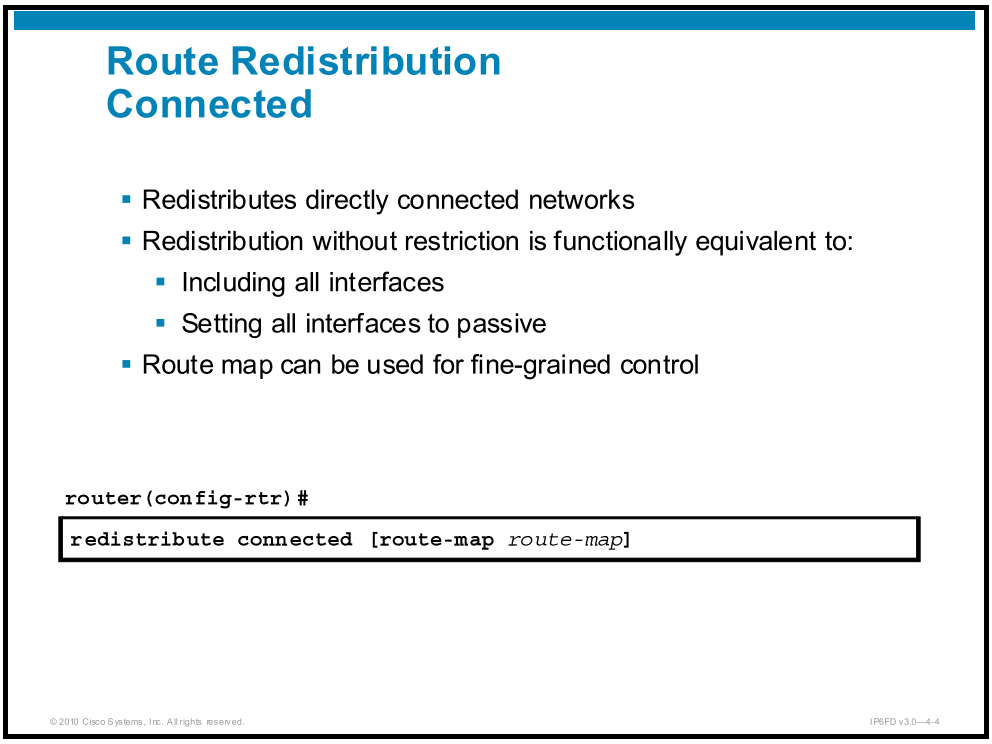
Route redistribution is needed in an environment where more than one routing protocol is used to convey reachability of a set of prefixes. The reasons for using heterogeneous routing environment may vary from device support to specific requirements by customers. Routing information is not automatically shared between two routing protocols. It must be configured explicitly with redistribution. When configuring redistribution, you may want to set the metric, because metrics in different routing protocols are incompatible.

IPv6 redistribution is conceptually the same as redistribution of IPv4 routes. However, due to some differences in configuration, redistribution of IPv6 information works slightly differently.

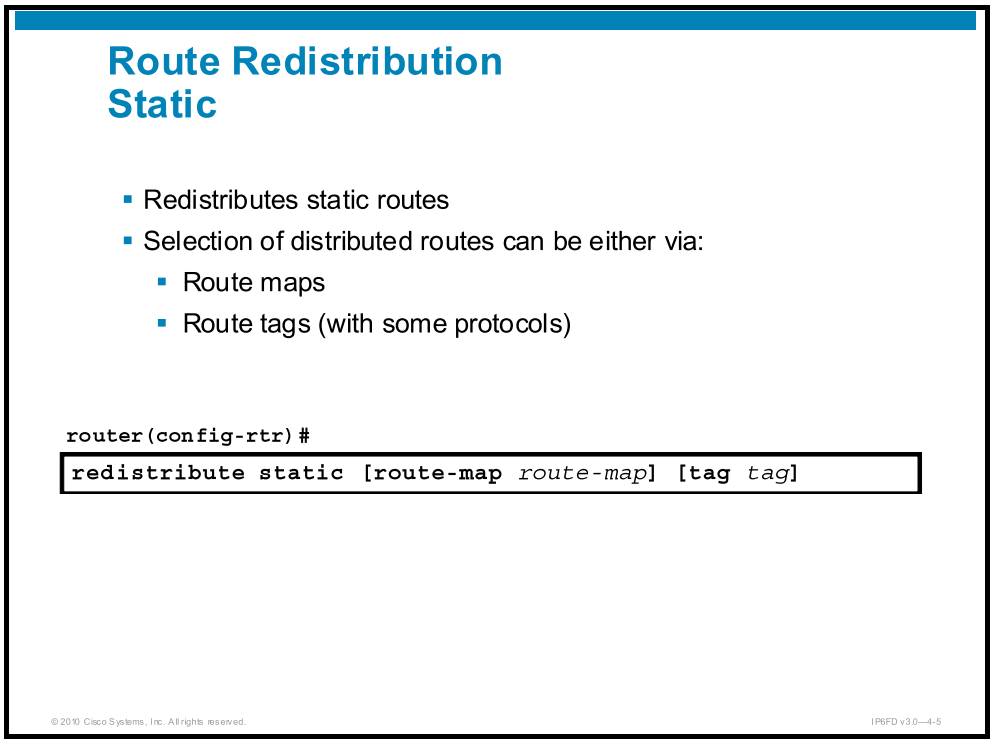
In IPv6 routing, there are no **network** commands in routing protocol configurations for inclusion of IPv6 interfaces. The exception to this rule is Border Gateway Protocol (BGP), but network command works differently in BGP.

As a consequence, the feature of Enhanced Interior Gateway Routing Protocol (EIGRP), where a static route pointed to an interface covered by a network command was included in the routing process, does not work for IPv6 routes.

Redistribution also does not include directly connected segments, even if they are covered by an interior gateway protocol (IGP) and are seen in IGP on other routers. This behavior differs from the behavior of IPv4.

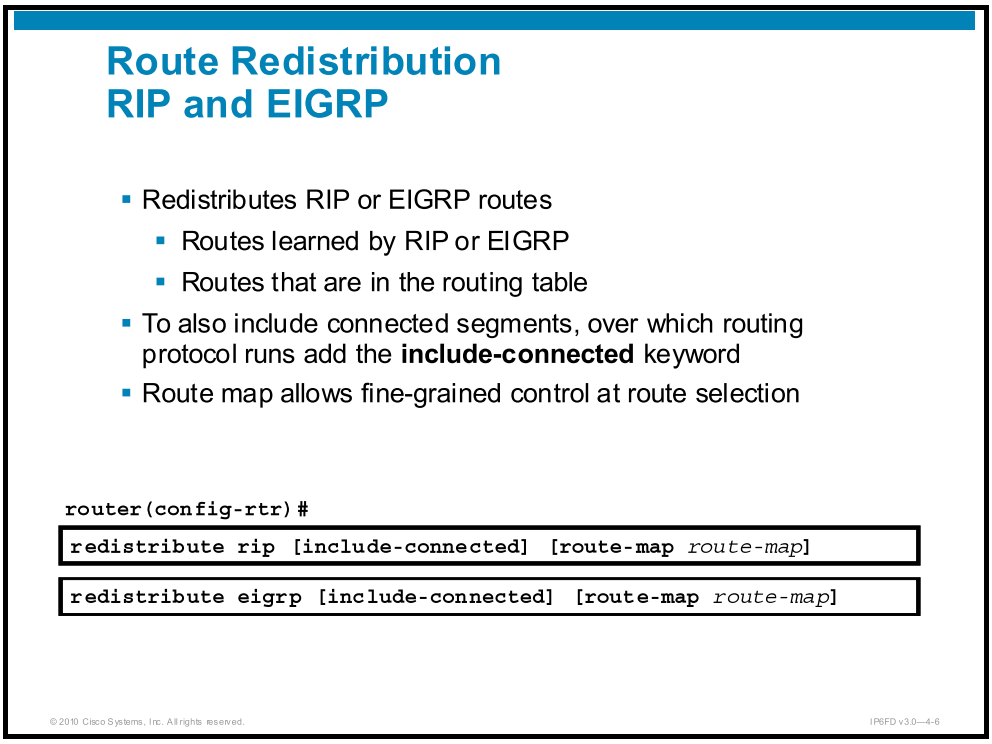


Connected segments can be included in a routing protocol in two ways. The first is by activating a routing protocol on the interface. The second is by redistributing connected segments. The first method is necessary if you also want to establish neighborships on that interface. If you do not, the recommended practice is to set the interface into passive mode. The combination of inclusion in the routing process and setting of passive mode on the interface is effectively the same as redistribution of connected routes. The difference lies in the way of controlling the selection of segments. With the first approach, you control inclusion by enabling or disabling the routing protocol on an interface. With the second approach, you must use a route map to select redistributed segments.



Static routing is the preferred method where networks are connected with single links. The lack of redundant links removes the requirement for path recalculation. Static routing is far less complex than dynamic routing and is available on all platforms. Where the part of the network that uses static routing touches the dynamically routed network, it is necessary to redistribute static routes into a routing protocol.

Redistribution of static routes can be controlled with route maps, like connected routes. In addition, some protocols (Open Shortest Path First [OSPF] and Intermediate System-to-Intermediate System [IS-IS]) allow selection of redistributed routes by tag value, directly. Other protocols can also achieve this result, but you need to use route maps.



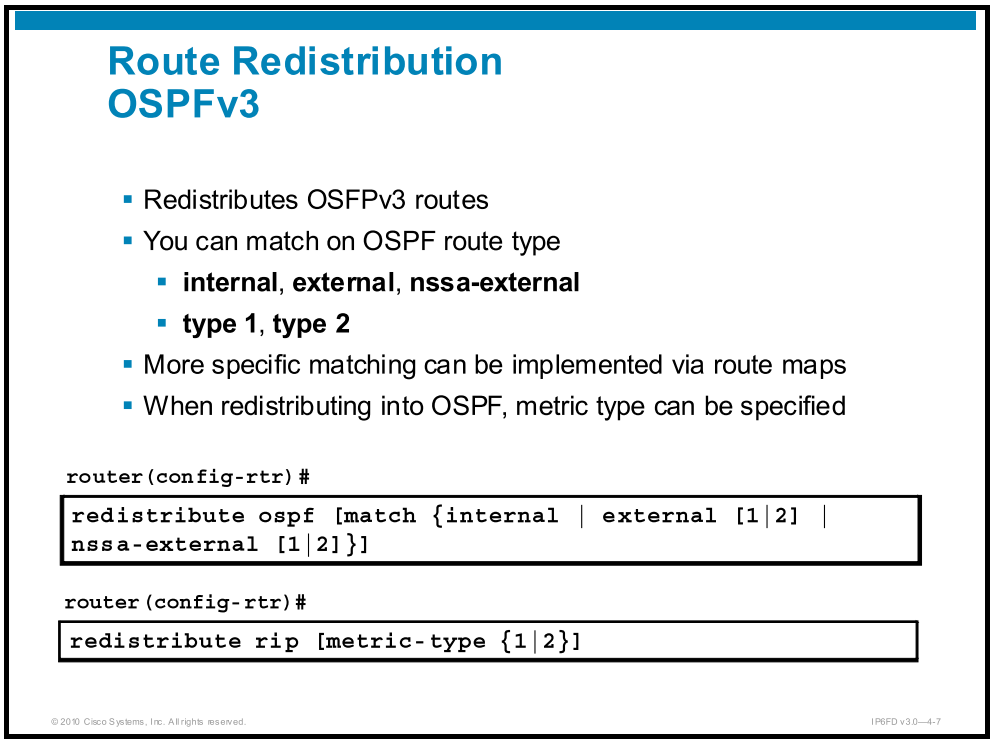
Routing Information Protocol (RIP) and EIGRP are both distance vector protocols and have similar redistribution commands. When RIP or EIGRP routes are redistributed, the router looks in a routing table for all entries that are learned by the protocol being redistributed. You can see these routes by using the **show ipv6 route rip** or **show ipv6 route eigrp** commands. The redistribution process does not include connected segments that are covered by RIP or EIGRP. To include these segments, add the **include-connected** parameter. As with the previous redistribution, you can achieve fine-grained control with the use of route maps.

When redistributing into RIP or EIGRP, you can specify the metric of redistributed routes.

With RIP, you need to specify the number of hops with the **metric** *hops* parameter when configuring redistribution. By default, redistributed routes will have the hop count set to 16, making them unreachable.

Note	Setting the metric to 15 at the point of redistribution will make the metric on the neighboring router 16, which will make it unreachable. The highest value that is useful is thus 14. Setting the redistribution metric to 14 will allow the routes to be propagated one hop from the redistributing router.

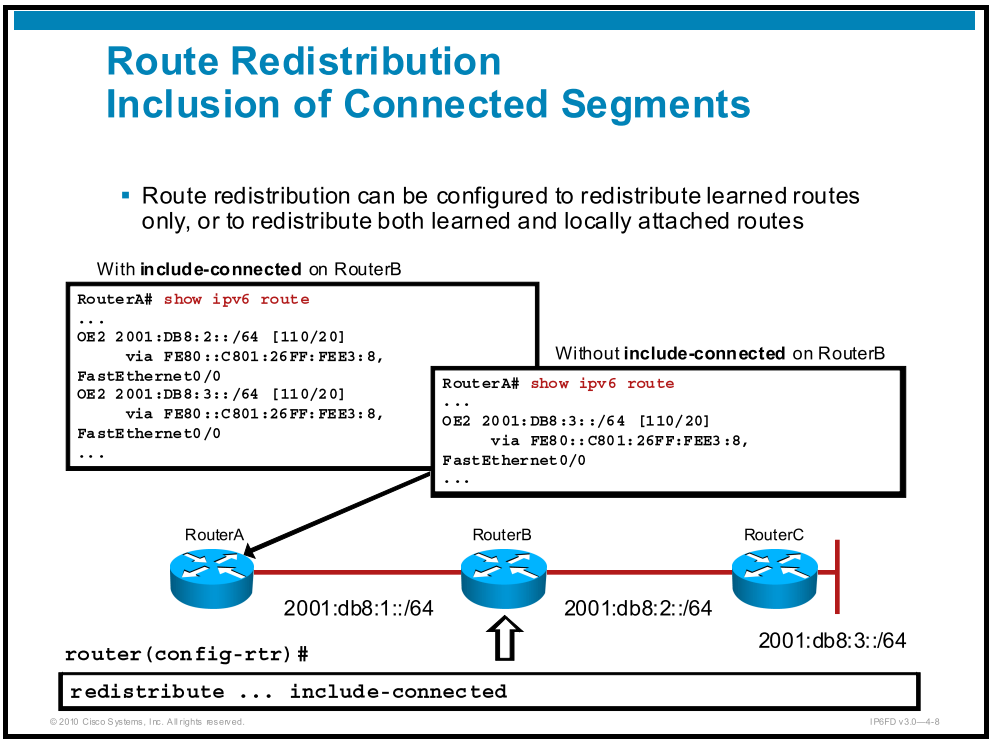
With EIGRP, you have two options. You can specify the metric as you can with RIP when you configure redistribution. With this approach, you can set different metrics for different routing protocols. The second option is to specify a default metric of redistributed routes with the **default-metric** command. This option is, however, not necessary for redistribution of connected routes, which have the metric set to zero by default.



OSPF version 3 (OSPFv3), being a link state protocol, adds a few additional options to all the options of RIP and EIGRP. When redistributing OSPFv3, you can match directly on route type. Options are the following:

- Internal routes
- External routes:
  - All external routes
  - Type 1 external routes
  - Type 2 external routes
- Not-so-stubby area (NSSA) external routes:
  - All NSSA external routes
  - Type 1 NSSA external routes
  - Type 2 NSSA external routes

When redistributing into OSPFv3, you can specify metric type as well as cost. The type can be either 1 or 2. Type 1 external routes are managed in the same way as any internal route. On every hop, the cost of exit interface is added to the existing hop of the route. The cost of type 2 external routes is not modified inside OSPF domain. It remains the same. Type 2 is the default type.



This figure gives an example of Cisco IOS Software behavior when redistributing regarding connected segments.

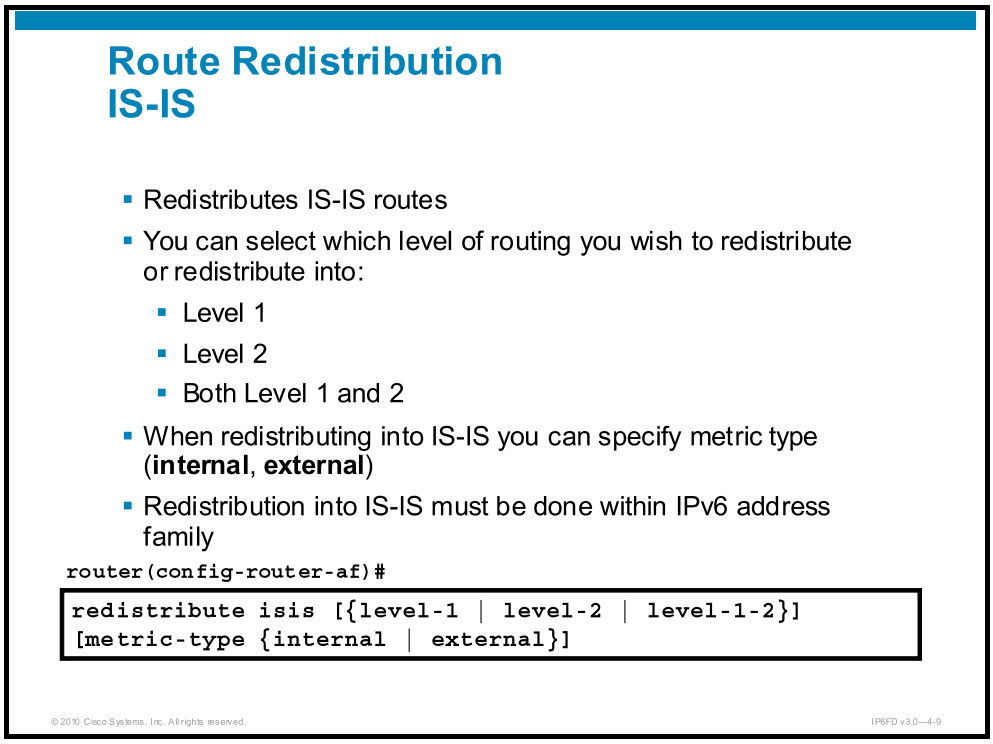
Redistribution into OSPF is configured on RouterB. The figure shows the output of the **show ipv6 route** command on RouterA. If the **include-connected** keyword is used on RouterB, you can see the presence of segment 2001:db8:2::/64 in the routing table. This behavior is the default in IPv4, but not in IPv6. The default IPv6 redistribution result can be seen on the right side of the figure, where you see only the LAN segment behind RouterC.

The default behavior is preferred in service provider environments, where transit segments are usually of no importance to the end customer. If your network requires these segments to be present in the routing table, you can get IPv4-like behavior by adding the **include-connected** keyword to the redistribution command when redistributing any dynamic routing protocol.

Note	The IPv6 routing infrastructure can be set with only link-local addressing on the links between the routers. This configuration increases security since it is not possible to attack a router at its interface address unless the attacker is present on the local link. However, it inhibits troubleshooting because tools such as traceroute might not work correctly.

Requirements for routes to be redistributed are as follows:

- The subnet is directly connected on the router.
- The interface is included in the routing process that is being redistributed.



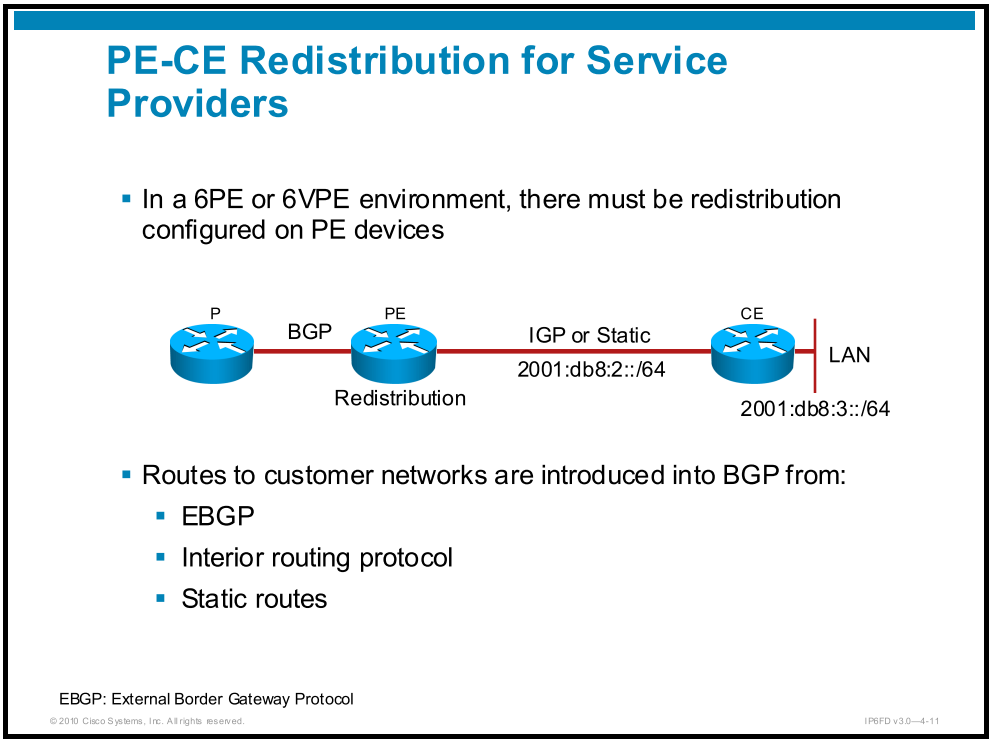
IS-IS is a hierarchical link-state routing protocol, like OSPF. You have all the options of other routing protocols; you can select redistributed routes via tags, enforce fine-grained control with route maps, and include connected segments, if desired.

Specifically, it is possible to control redistribution from and to different levels of IS-IS routing. IS-IS uses Level 2 routing instead of OSPF backbone area 0. Level 2 routing takes care of interarea routing. Level 1 routing routes traffic inside one area.

When redistributing IS-IS, you can specify whether to redistribute Level 1 routes, Level 2 routes, or both. You can also specify into which level you are redistributing. Additionally, you can set the metric type to either internal or external.

PE-CE Redistribution for Service Providers

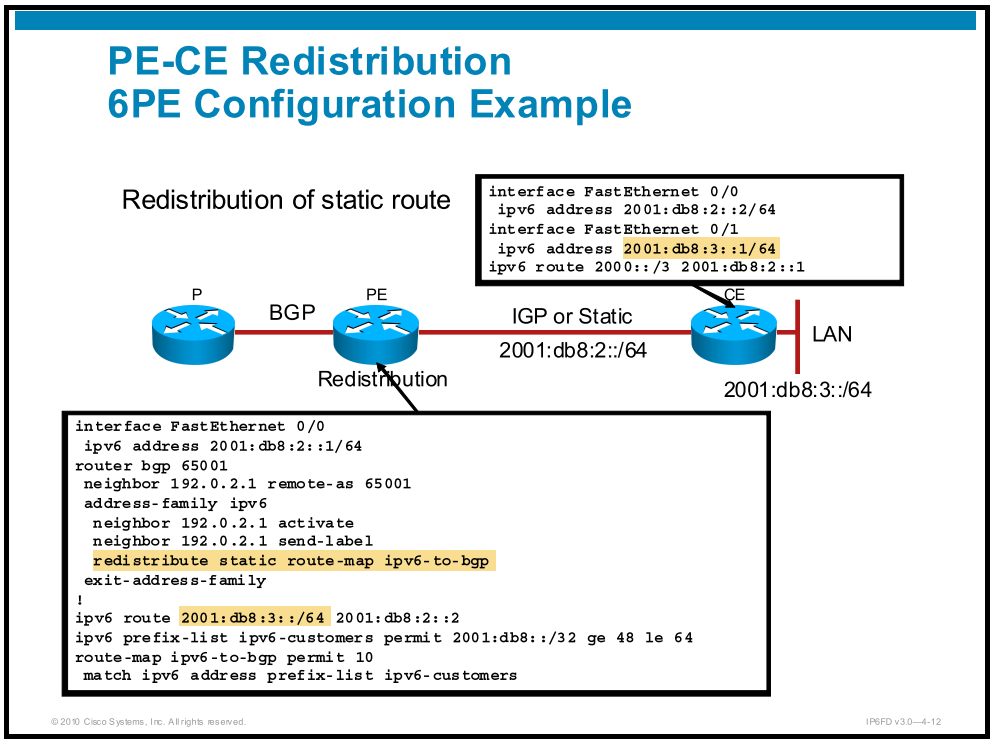
This topic describes provider edge to customer edge (PE-CE) redistribution for service providers.



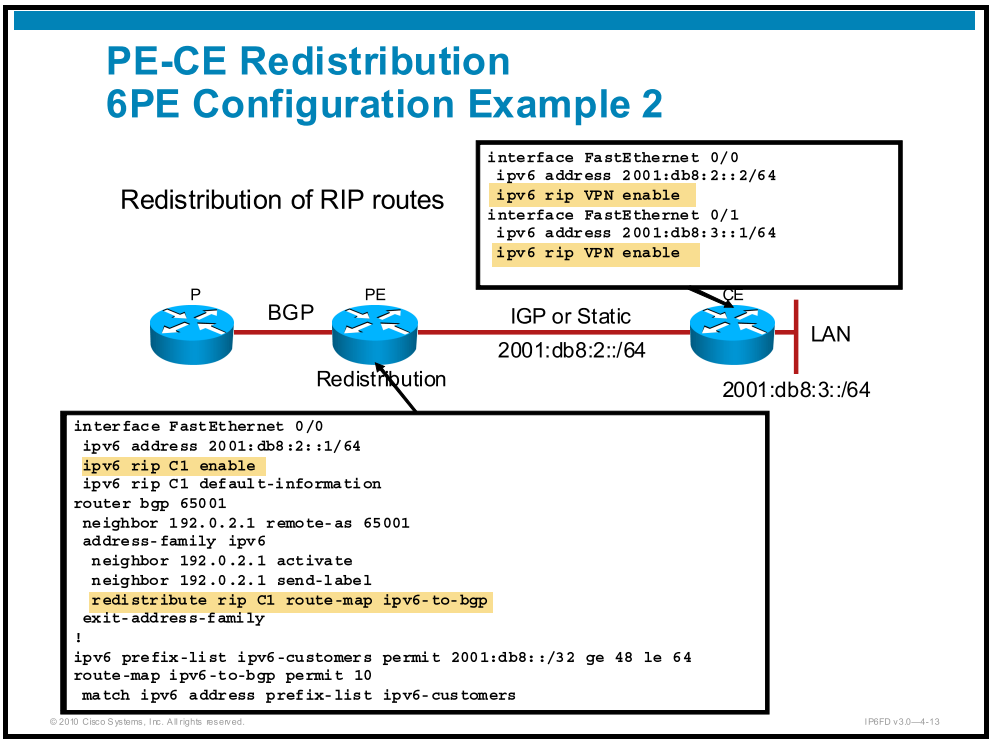
In a service provider environment, BGP is the routing protocol of choice because of its extreme scalability, compared to other routing protocols, and the level of control available.

For IPv6 connectivity over Multiprotocol Label Switching (MPLS), currently the available choices include Cisco IPv6 Provider Edge Router (6PE) and IPv6 VPN Provider Edge Router (6VPE) over MPLS.

In such scenarios, the routing inside the service provider network for customer prefixes will be provided by BGP. The routing between the PE and CE can be anything that the service provider and customer agree on. Most commonly, static routes are used for a simple single-homed deployment, with dynamic routing used when more links are used for redundancy.



The figure shows an example of static PE-CE routing. Static routes are redistributed on the PE router into BGP using a route map, which allows only prefixes assigned to customers. The customer has a default route that is configured for outbound traffic.

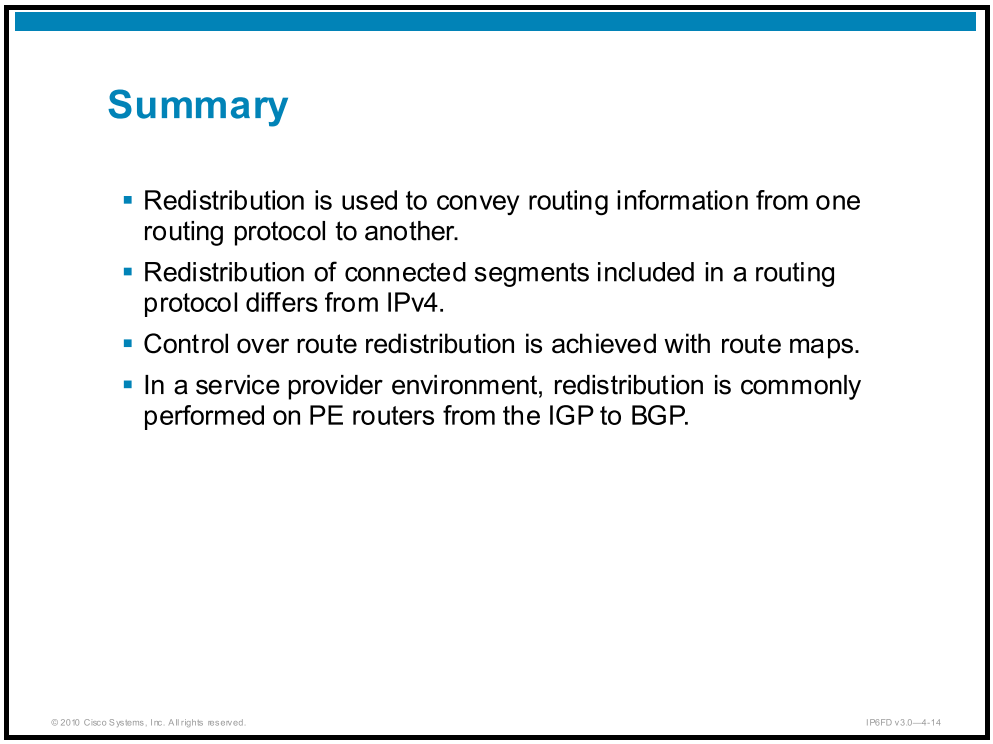


This figure shows an example of dynamic routing setup. The service provider and the customer are running RIP for IPv6 routing information exchange. The customer advertises the LAN segment, while the service provider advertises the default route.

RIP is redistributed into BGP so that other PE routers know the location of customer prefixes. Redistribution is done with a route map to safeguard against introduction of unwanted routes into BGP.

Summary

This topic summarizes the key points that were discussed in this lesson.



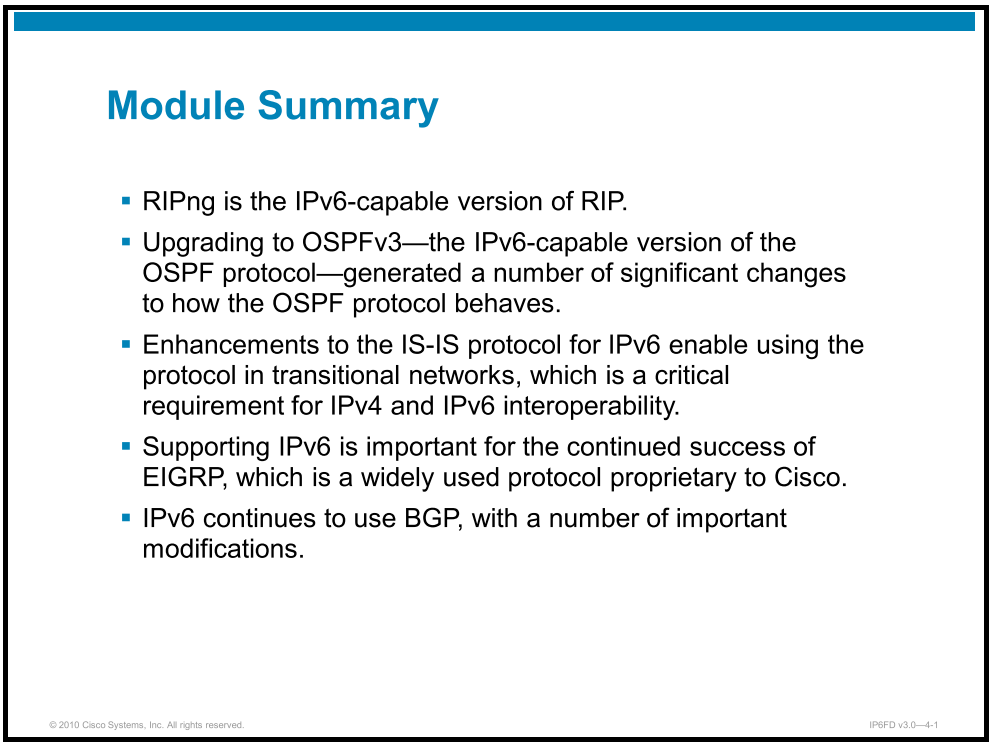
Resources

For additional information, refer to these resources:

- *Redistributing Routes into an IPv6 RIP Routing Process*
  http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-rip.html#wp1041727

Module Summary

This topic summarizes the key points that were discussed in this module.



Module Self-Check

Use the questions here to review what you learned in this module. The correct answers are found in the Module Self-Check Answer Key.

Q1) Which two mechanisms are the same in RIPv2 and RIPng? (Choose two.) (Source: Routing with RIPng)

A) distance vector algorithm
B) DUAL algorithm
C) maximum radius of 15 router hops end-to-end in the network
D) support for areas

Q2) Which commands enable RIP for IPv6 in a network? (Source: Routing with RIPng)

A)
```
interface FastEthernet 0/0
 ipv6 rip RIP5 enable
```
B)
```
ipv6 router rip RIP5
 network 192.0.2.0 netmask 0.0.0.255
```
C)
```
ipv6 router ripng
 network 192.0.2.0/24
```
D)
```
interface FastEthernet 0/0
 network 192.0.2.0 netmask 0.0.0.255
```

Q3) Which item represents similarities between OSPFv3 and OSPFv2? (Source: Examining OSPFv3)

A) support for IPv4 and IPv6
B) support for IPv4 in the case of OSPFv2 and multiprotocol support for OSPFv3
C) enabled per-link, rather than per-network, using network statements
D) link-state routing protocols

Q4) Which of these features was removed from OSPFv2 for OSPFv3? (Source: Examining OSPFv3)

A) area summarization
B) distance vector optimization process
C) authentication
D) periodic update processing

Q5) What does the term “single SPF” mean? (Source: Examining Integrated IS-IS)

A) There is a single routing database for IPv4 and IPv6.
B) There are separate routing databases for IS-IS, but a single routing table.
C) The IPv4 and IPv6 topology must be identical through the entire routing domain.
D) The IPv4 and IPv6 topology must be congruent throughout an IS-IS area (Layer 1 links).

Q6) Under which submode are IPv6-specific IS-IS attributes configured? (Source: Examining Integrated IS-IS)

A) address-family ipv6
B) interface FastEthernet0/0
C) ipv6 router isis TAG1
D) is-is router TAG1

Q7) Which term describes generic EIGRP support for multiple protocols? (Source: Examining EIGRP for IPv6)

A) PIM6
B) EIGRP multiprotocol
C) EIGRP multitopology
D) protocol-dependent modules

Q8) Which command does EIGRP have that no other routing protocol has? (Source: Examining EIGRP for IPv6)

A) **enable multilevel**
B) **routing eigrp enable**
C) **shutdown**
D) **begin route redistribution**

Q9) Which two new attributes support IPv6 in MP-BGP (BGP4+)? (Choose two.) (Source: Understanding MP-BGP)

A) multiprotocol destination
B) multiprotocol feasible
C) multiprotocol network
D) multiprotocol next hop
E) multiprotocol reachable
F) multiprotocol subnet
G) multiprotocol successor
H) multiprotocol unreachable

Q10) BGP peering over IPv6 can be done using link-local addresses under which condition? (Source: Understanding MP-BGP)

A) The peers are IBGP only; EBGP peering cannot be done over link-local addresses.
B) The peers are within the same AS.
C) The peers have interfaces on the same subnet.
D) The peers have interfaces within the same site boundary.

Q11) Which two tasks is BGP-prefix filtering used for? (Choose two.) (Source: Understanding MP-BGP)

A) filtering prefixes before announcing them to BGP peers
B) filtering prefixes before accepting announcements from BGP peers
C) filtering prefixes after accepting them locally but before passing them to IBGP peers
D) filtering prefixes before redistributing them into an IGP

Module Self-Check Answer Key

Q1) A, C

Q2) A

Q3) D

Q4) C

Q5) D

Q6) A

Q7) D

Q8) C

Q9) E, H

Q10) C

Q11) A, B