BSCI

Building Scalable Cisco Internetworks

Volume 2

Version 3.0

Student Guide

Editorial, Production, and Graphic Services: 06.14.06

Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

European Headquarters
Cisco Systems International BV
Haarlerbergpark
Haarlerbergweg 13-19
1101 CH Amsterdam
The Netherlands
www-europe.cisco.com
Tel: 31 0 20 357 1000
Fax: 31 0 20 357 1100

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
www.cisco.com
Tel: 408 526-7660
Fax: 408 527-0883

Asia Pacific Headquarters
Cisco Systems, Inc.
168 Robinson Road
#28-01 Capital Tower
Singapore 068912
www.cisco.com
Tel: +65 6317 7777
Fax: +65 6317 7799

Cisco Systems has more than 200 offices in the following countries and regions. Addresses, phone numbers, and fax numbers are listed on the **Cisco.comWebsiteatwww.cisco.com/go/offices.**

Argentina • Australia • Austria • Belgium • Brazil • Bulgaria • Canada • Chile • China PRC • Colombia • Costa Rica • Croatia • Cyprus • Czech Republic • Denmark • Dubai, UAE • Finland • France • Germany • Greece • Hong Kong SAR • Hungary • India • Indonesia • Ireland Israel • Italy • Japan • Korea • Luxembourg • Malaysia • Mexico • The Netherlands • New Zealand • Norway • Peru • Philippines Poland • Portugal • Puerto Rico • Romania • Russia • Saudi Arabia • Scotland • Singapore • Slovakia • Slovenia • South Africa Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • United States • Venezuela • Vietnam • Zimbabwe

DISCLAIMER WARRANTY: THIS CONTENT IS BEING PROVIDED "AS IS." CISCO MAKES AND YOU RECEIVE NO WARRANTIES IN CONNECTION WITH THE CONTENT PROVIDED HEREUNDER, EXPRESS, IMPLIED, STATUTORY OR IN ANY OTHER PROVISION OF THIS CONTENT OR COMMUNICATION BETWEEN CISCO AND YOU. CISCO SPECIFICALLY DISCLAIMS ALL IMPLIED WARRANTIES, INCLUDING WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. This learning product may contain early release content, and while Cisco believes it to be accurate, it falls subject to the disclaimer above.

Table of Contents

Volume 2

Module 5

Manipulating Routing Updates

Overview

This module explains why it is necessary to manipulate routing information. During route redistribution between IP routing domains, suboptimal routing can occur without manipulation. There are also times when routing information would waste bandwidth on a router interface because routing information is not needed.

This module provides a description and examples of methods to implement the controls described above with Cisco Systems devices.

Module Objectives

Upon completing this module, you will be able to manipulate routing and packet flow. This ability includes being able to meet these objectives:

- Explain what route distribution is and why it may be necessary
- Configure route redistribution between multiple IP routing protocols
- Configure dynamic routing protocol updates for passive interfaces and distribute lists
- Describe and configure DHCP services

Lesson 1

Operating a Network Using Multiple IP Routing Protocols

Overview

Simple routing protocols work well for simple networks, but as networks grow and become more complex, it may be necessary to change routing protocols. Often the transition between routing protocols takes place gradually, so there are multiple routing protocols that are operating in the network for variable lengths of time. This lesson examines several reasons for using more than one routing protocol.

It is important to understand how to exchange routing information between these routing protocols and how Cisco routers operate in a multiple routing-protocol environment. This lesson describes migration from one routing protocol to another and how Cisco routers make route selections when multiple protocols are active in the network.

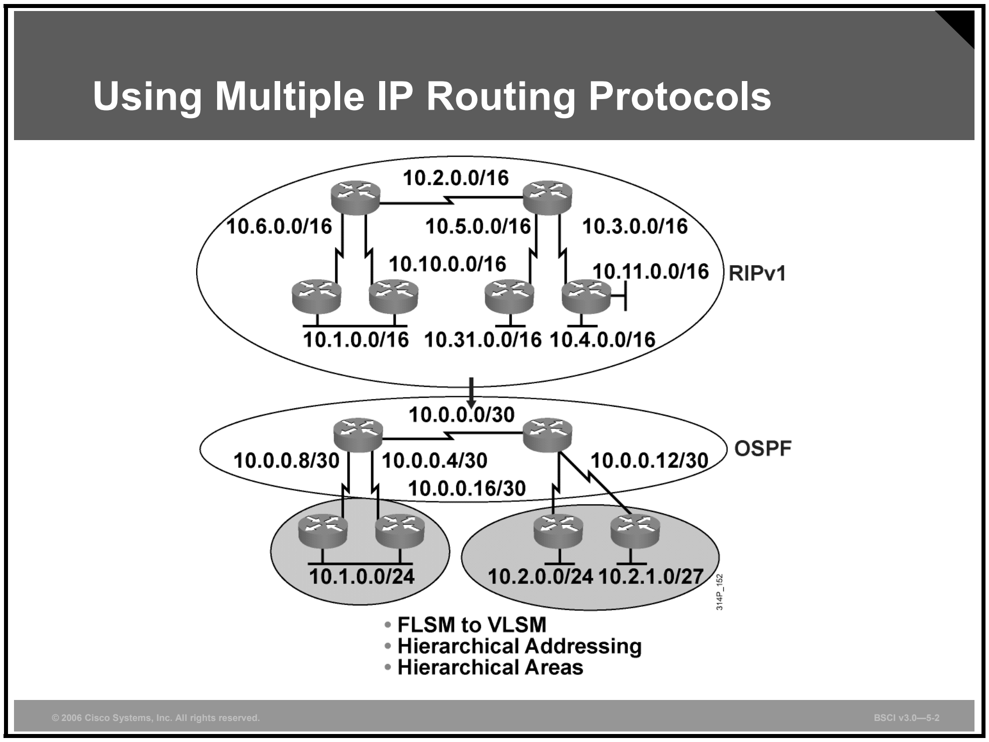
Objectives

Upon completing this lesson, you will be able to explain what route distribution is and why it may be necessary. This ability includes being able to meet these objectives:

- Explain the need to use multiple IP routing protocols
- Define route redistribution
- Identify the seed metrics that are used by various routing protocols

Using Multiple IP Routing Protocols

This topic describes the issues related to migrating from one routing protocol to another.



Using Multiple IP Routing Protocols

- FLSM to VLSM
- Hierarchical Addressing
- Hierarchical Areas

There are many reasons why a change in routing protocols may be required. For example, as a network grows and becomes more complex, the original routing protocol may no longer be the best choice. Remember that Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP) periodically send their entire routing tables in their updates.

As the network grows larger, the traffic from those updates can slow the network down, indicating that a change to a more scalable routing protocol may be necessary. Alternatively, perhaps you are using IGRP or Enhanced IGRP (EIGRP) and need a protocol that supports multiple vendors or your company implements a policy that specifies a particular routing protocol.

Whatever the reason for the change, network administrators must conduct migration from one routing protocol to another carefully and thoughtfully. The new routing protocol will most likely have requirements and capabilities that are different from the old one.

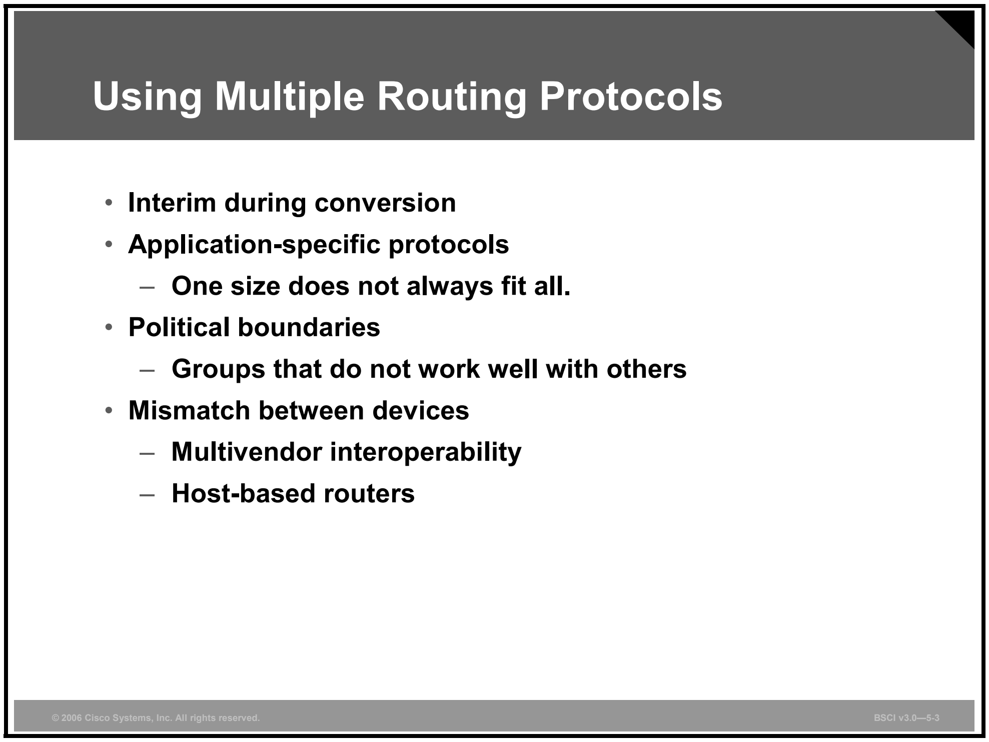
It is important for network administrators to understand what must be changed and to create a detailed plan before making any changes. An accurate topology map of the network and an inventory of all network devices are also critical for success.

Link-state routing protocols, such as Open Shortest Path First (OSPF) and Intermediate System-to-Intermediate System (IS-IS), require a hierarchical network structure. Network administrators need to decide which routers will reside in the backbone area and how to divide the other routers into areas. While EIGRP does not require a hierarchical structure, it operates much more effectively within one.

During the transition, there will likely be a time when both routing protocols are running in the network, which may require redistribution of routing information between the two protocols. If so, carefully plan the redistribution strategy to avoid disrupting network traffic or causing outages.

Defining Route Redistribution

This topic describes the purpose of route redistribution.



Using Multiple Routing Protocols

- **Interim during conversion**
- **Application-specific protocols**
  - **One size does not always fit all.**
- **Political boundaries**
  - **Groups that do not work well with others**
- **Mismatch between devices**
  - **Multivendor interoperability**
  - **Host-based routers**

Multiple routing protocols may be necessary in the following situations:

- When you are migrating from an older interior gateway protocol (IGP) to a new IGP, multiple routing protocols are necessary. Multiple redistribution boundaries may exist until the new protocol has completely displaced the old protocol.
- When use of another protocol is desired, but the old routing protocol is needed for host systems, multiple routing protocols are necessary, for example, UNIX host-based routers running RIP.
- Some departments might not want to upgrade their routers to support a new routing protocol.
- In a mixed-router vendor environment, you can use a routing protocol specific to Cisco such as EIGRP in the Cisco portion of the network and a common standards-based routing protocol, like OSPF, to communicate with devices from other vendors.

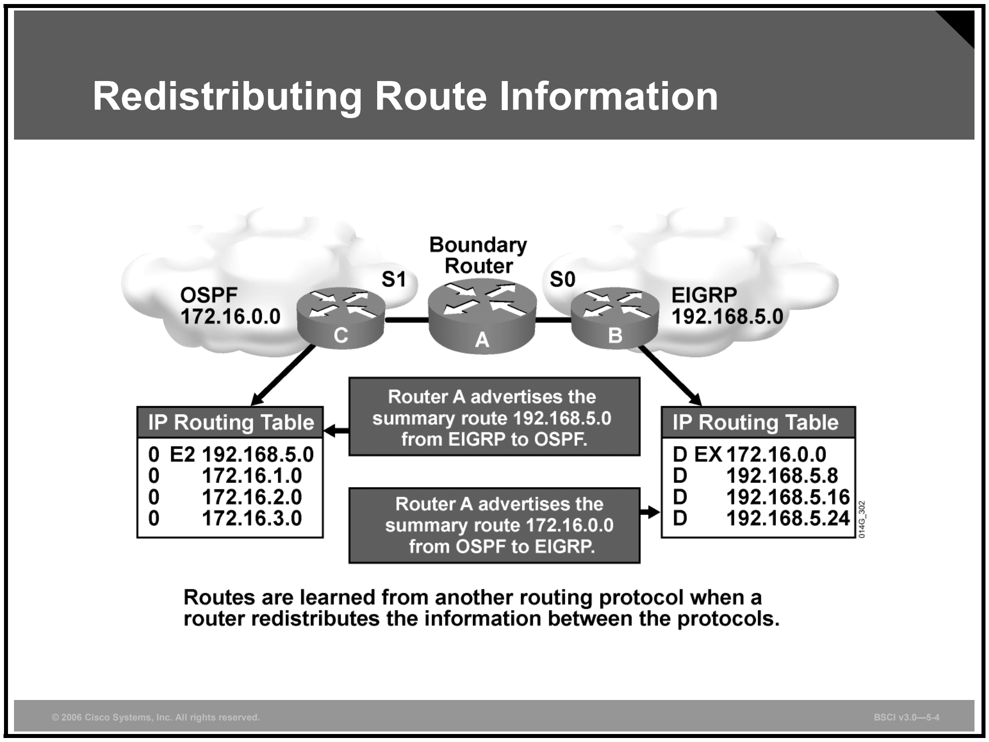
When multiple routing protocols are running in different parts of the network, there may be a need for hosts in one part of the network to reach hosts in the other part. One solution is to advertise a default route into each routing protocol, but that is not always the best policy. The network design may not allow default routes.

If there is more than one way to get to a destination network, routers may need information about routes in the other parts of the network to determine the best path to that destination. Additionally, if there are multiple paths, a router must have sufficient information to determine a loop-free path to the remote networks.

Cisco routers allow internetworks using different routing protocols, referred to as routing domains or autonomous systems, to exchange routing information through a feature called route redistribution.

Redistribution is how routers connect different routing domains so that they can exchange and advertise routing information between the different autonomous systems.

Note The term autonomous system (AS), as used here, denotes internetworks using different routing protocols. These routing protocols may be IGPs or exterior gateway protocols (EGPs), which is a different use of the term “AS” than when in Border Gateway Protocol (BGP).



Redistributing Route Information

IP Routing Table	
0 E2	192.168.5.0
0	172.16.1.0
0	172.16.2.0
0	172.16.3.0

Router A advertises the summary route 192.168.5.0 from EIGRP to OSPF.

Router A advertises the summary route 172.16.0.0 from OSPF to EIGRP.

IP Routing Table	
D EX	172.16.0.0
D	192.168.5.8
D	192.168.5.16
D	192.168.5.24

Routes are learned from another routing protocol when a router redistributes the information between the protocols.

BSCI v3.0—5-4

Within each AS, the internal routers have complete knowledge about their network. The router that interconnects the autonomous systems is called a boundary router. The boundary router must be running all the routing protocols that will be exchanging routes.

In most cases, route redistribution must be configured in order to redistribute routes from one routing protocol to another routing protocol. The only time that redistribution is automatic in IP routing protocols is between IGRP and EIGRP processes running on the same router and using the same AS number.

When a router redistributes routes, it allows a routing protocol to advertise routes that were not learned through that routing protocol. These redistributed routes could have been learned via a different routing protocol, such as when redistributing between EIGRP and OSPF, and they also could have been learned from static routes or by a direct connection to a network.

Routers can redistribute static and connected routes, as well as routes from other routing protocols.

Redistribution is always performed outbound. The router doing redistribution does not change its routing table. When, for instance, redistribution between OSPF and EIGRP is configured, the OSPF process on the boundary router takes the EIGRP routes in the routing table and advertises them as OSPF routes to its OSPF neighbors.

Likewise, the EIGRP process on the boundary router takes the OSPF routes in the routing table and advertises them as EIGRP routes to its EIGRP neighbors. Then both autonomous systems will know about the routes of the other, and each AS can then make informed routing decisions for these networks.

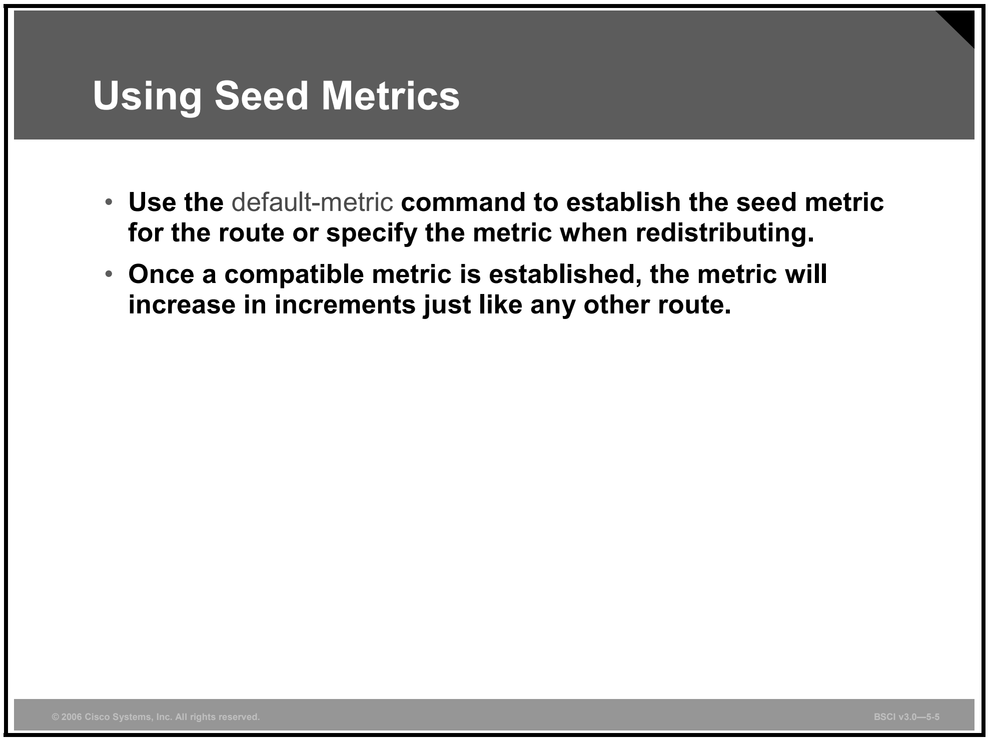
EIGRP neighbors use the EIGRP external (D EX) listing to route traffic destined for the other AS via the boundary router. The boundary router must have the OSPF routes for that destination network in its routing table to be able to forward the traffic.

For this reason, routes must be in the routing table for them to be redistributed. This requirement may seem self-evident, but it can also be a source of confusion.

For instance, if a router learns about a network via EIGRP and OSPF, only the EIGRP route is put in the routing table because it has a lower administrative distance. Suppose RIP is also running on this router, and you want to redistribute OSPF routes into RIP. That network will not be redistributed into RIP because it is in the routing table as an EIGRP route, not as an OSPF route.

Using Seed Metrics

This topic describes the seed metrics that are used by different routing protocols, as well as how and why to use seed metrics.



Using Seed Metrics

- **Use the** default-metric **command to establish the seed metric for the route or specify the metric when redistributing.**
- **Once a compatible metric is established, the metric will increase in increments just like any other route.**

BSCI v3.0—5-5

Each routing protocol defines a metric for each route. The metric value determines the shortest or “best” part to an IP network. When a router redistributes routes from one routing domain to another, this information cannot be translated from one routing protocol to another. For example, a RIP hop cannot be dynamically recalculated to an OSPF cost by the router doing redistribution.

Therefore, a seed metric is used to artificially set the distance, cost, and so on, to each external (redistributed) network from the redistribution point.

Seed Metrics Example

For example, if a boundary router receives a RIP route, the route will have hop count as a metric. To redistribute the route into OSPF, the router must translate the hop count into a cost metric that the OSPF routers understand.

This seed metric, also referred to as the default metric, is defined during redistribution configuration. When the seed metric for a redistributed route is established, the metric increases in increments normally within the AS.

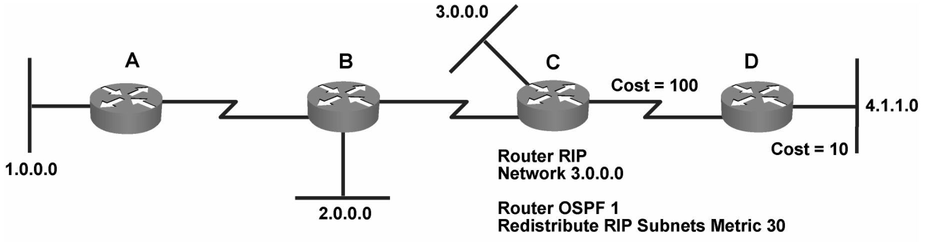
Note The exception to this rule is OSPF E2 routes, which hold their initial metric regardless of how far they are propagated across an AS.

The **default-metric** command, used in the routing process configuration mode, establishes the seed metric for all redistributed routes.

Cisco routers also allow the seed metric to be specified as part of the **redistribution** command, either with the **metric** option or by using a route map.

Whichever way it is done, the initial seed metric should be set to a value larger than the largest metric within the receiving AS to help prevent suboptimal routing and routing loops.

Redistribution with Seed Metric



Router RIP
Network 3.0.0.0

Router OSPF 1
Redistribute RIP Subnets Metric 30

Table A

```
C 1.0.0.0
R 120/1 2.0.0.0
R 120/2 3.0.0.0
```

Table B

```
C 2.0.0.0
R 120/1 1.0.0.0
R 120/1 3.0.0.0
```

Table C

```
C 3.0.0.0
R 120/1   2.0.0.0
R 120/2   1.0.0.0
O 110/110 4.1.1.0
```

Table D

```
C  4.1.1.0/24
E2 110/130 1.0.0.0
E2 110/130 2.0.0.0
E2 110/130 3.0.0.0
```

314P_155

The table lists protocol names with the default seed metrics for the various protocols.

Protocol	Default Seed Metrics
RIP	Infinity
IGRP or EIGRP	Infinity
OSPF	20 for all except BGP, which is 1
IS-IS	0
BGP	BGP metric is set to IGP metric value

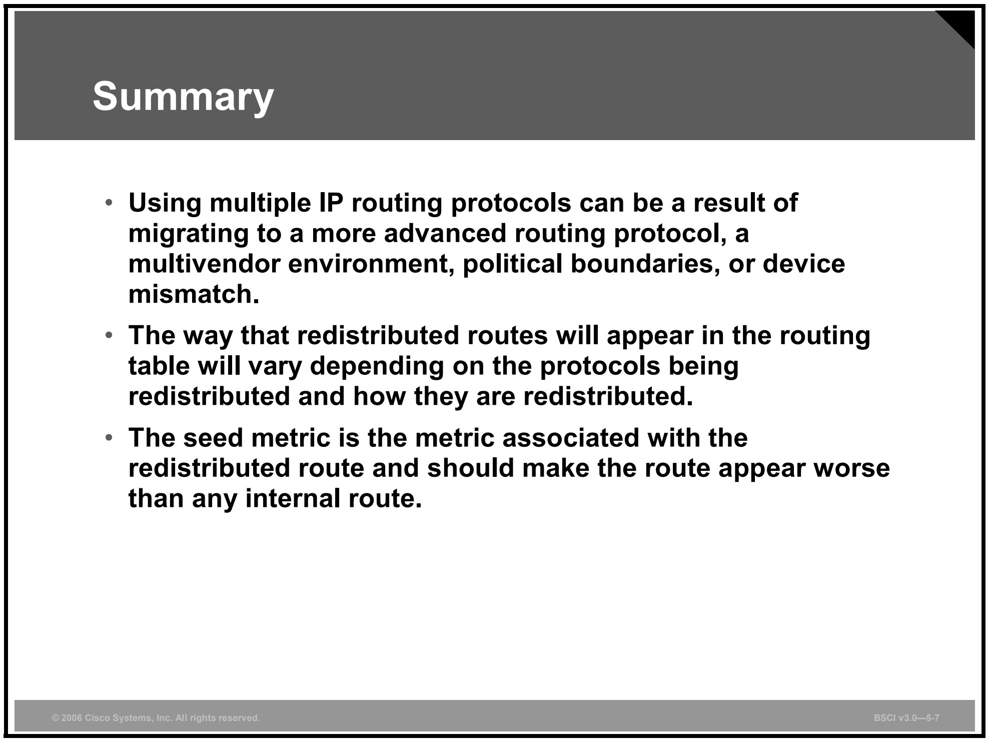
Default Seed Metrics Example

The figure illustrates a seed metric of 30 implemented by OSPF on the redistributed RIP routes. The link cost of the Ethernet link to router D is 100. So, the cost for networks 1.0.0.0, 2.0.0.0, and 3.0.0.0 in router D is the seed metric (30) plus the link cost (100) = 130. Notice that the metrics of the three networks in the RIP cloud is irrelevant in the OSPF cloud, because the objective is to have each OSPF router forward traffic for the three networks to the border (redistributing) router.

A metric of infinity tells the router that the route is unreachable, and therefore, it should not be advertised. When redistributing routes into RIP, IGRP, and EIGRP, you must specify a default metric. For OSPF, the redistributed routes have a default type 2 metric of 20, except for redistributed BGP routes, which have a default type 2 metric of 1. For IS-IS, the redistributed routes have a default metric of 0. But unlike RIP, IGRP, or EIGRP, a seed metric of 0 will not be treated as unreachable by IS-IS. Configuring a seed metric for redistribution into IS-IS is recommended. For BGP, the redistributed routes maintain the IGP routing metrics.

Summary

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



Summary

- Using multiple IP routing protocols can be a result of migrating to a more advanced routing protocol, a multivendor environment, political boundaries, or device mismatch.
- The way that redistributed routes will appear in the routing table will vary depending on the protocols being redistributed and how they are redistributed.
- The seed metric is the metric associated with the redistributed route and should make the route appear worse than any internal route.

Lesson 2

Configuring and Verifying Route Redistribution

Overview

Configuring route redistribution can be simple or complex, depending upon the mix of routing protocols that you want to redistribute. The commands that are used to enable redistribution and to assign metrics vary slightly depending upon the routing protocols being redistributed. Before configuring the exchange of routing information between routing protocols, you must understand the procedures for and requirements of each routing protocol.

Redistribution must be configured correctly for each routing protocol to obtain proper results. This lesson describes how to configure route redistribution between various IGP (interior gateway protocol) routing protocols. The commands for each protocol are covered. These commands differ slightly, according to the different routing protocol requirements. In addition, the impact of route redistribution is analyzed.

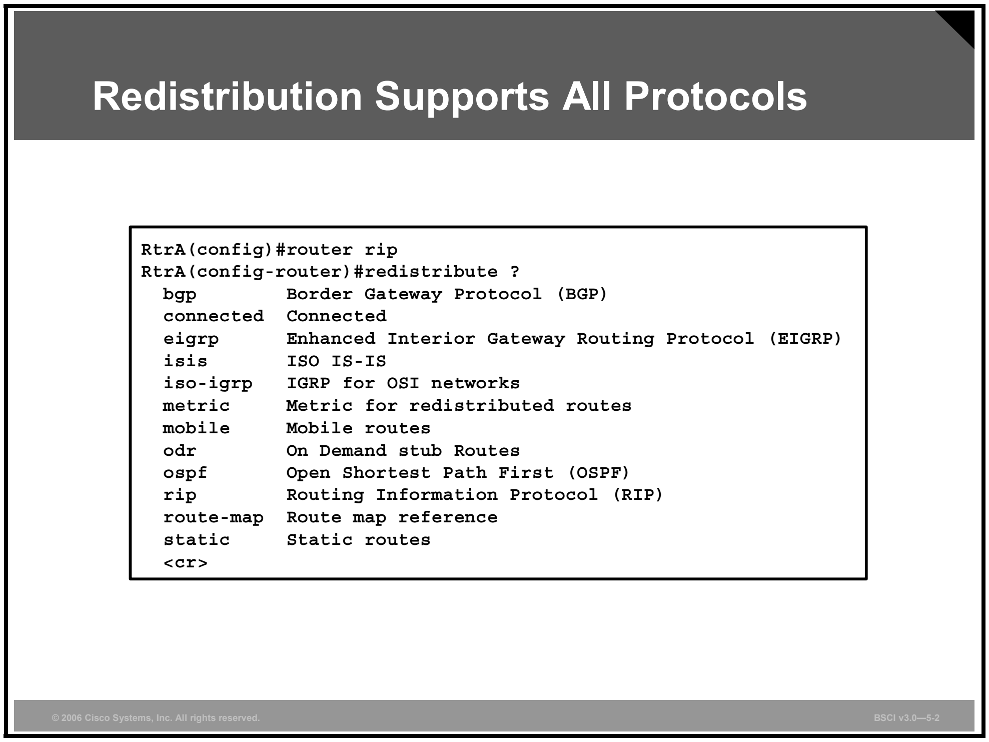
Objectives

Upon completing this lesson, you will be able to configure route redistribution between multiple IP routing protocols. This ability includes being able to meet these objectives:

- Describe the steps necessary to configure route redistribution
- Describe how to redistribute routes into RIP
- Describe how to redistribute routes into OSPF
- Describe how to redistribute routes into EIGRP
- Describe how to redistribute routes into IS-IS
- Describe how to verify route redistribution operations

Configuring Redistribution

This topic describes how to configure route redistribution.



Redistribution Supports All Protocols

```
RtrA(config)#router rip
RtrA(config-router)#redistribute ?
  bgp        Border Gateway Protocol (BGP)
  connected  Connected
  eigrp      Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis       ISO IS-IS
  iso-igrp   IGRP for OSI networks
  metric     Metric for redistributed routes
  mobile     Mobile routes
  odr        On Demand stub Routes
  ospf       Open Shortest Path First (OSPF)
  rip        Routing Information Protocol (RIP)
  route-map  Route map reference
  static     Static routes
  <cr>
```

Example: Redistribution Supports All Protocols

As shown in the example in the figure, redistribution supports all routing protocols. Additionally, static and connected routes can be redistributed to allow the routing protocol to advertise the routes without using a network statement for them.

Routes are redistributed into a routing protocol, and so the **redistribute** command is given under the routing process that is to receive the routes. Before implementing redistribution, consider these points:

- Only protocols that support the same protocol stack are redistributed. For example, you can redistribute between IP Routing Information Protocol (RIP) and Open Shortest Path First Protocol (OSPF) because they both support the TCP/IP stack.

  You cannot redistribute between Internetwork Packet Exchange (IPX) RIP and OSPF because IPX RIP supports the IPX/Sequenced Packet Exchange (SPX) stack and OSPF does not. Although there are different protocol-dependent modules of Enhanced Interior Gateway Routing Protocol (EIGRP) for IP, IPX, and AppleTalk, routes cannot be redistributed between them because each protocol-dependent module (PDM) supports a different protocol stack.

- The method used to configure redistribution varies slightly among different routing protocols and combinations of routing protocols. For example, redistribution occurs automatically between Interior Gateway Routing Protocol (IGRP) and EIGRP when they have the same autonomous system (AS) number; however, redistribution must be configured between all other routing protocols. Some routing protocols require a metric to be configured during redistribution, but others do not.

Note IGRP is no longer supported, as of Cisco IOS Software Release 12.3.

The following generic steps apply to all routing protocol combinations; however, the commands that are used to implement these steps may vary. For configuration commands, it is important that you review the Cisco IOS documentation for the specific routing protocols that need to be redistributed.

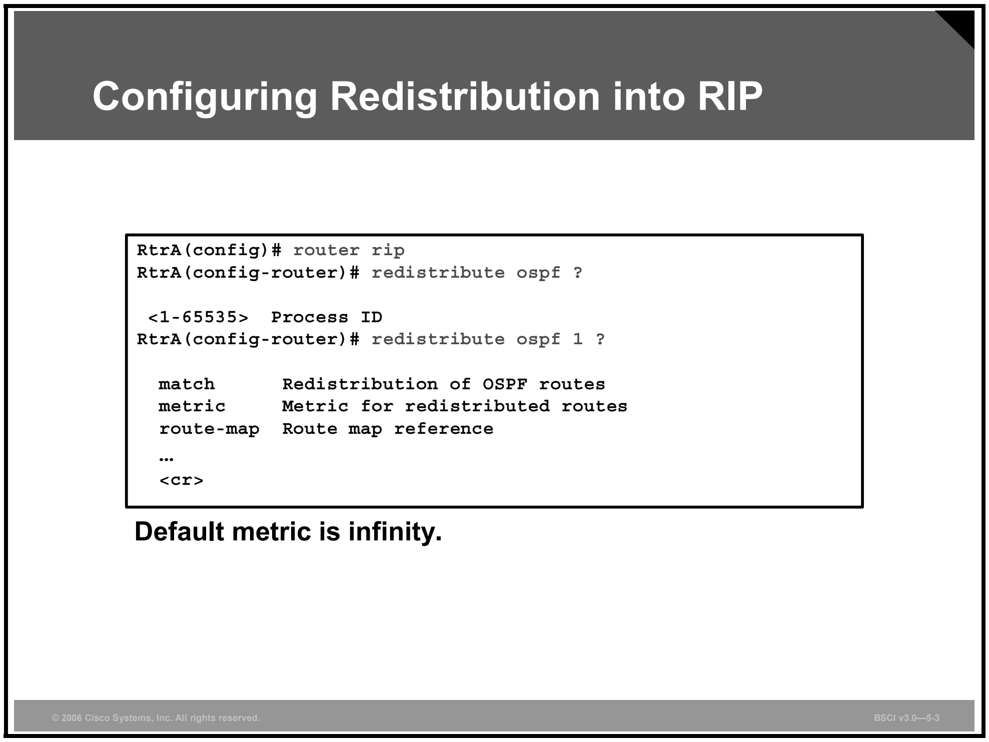
Note In this topic, the terms "core" and "edge" are generic terms that are used to simplify the discussion about redistribution.

1. Locate the boundary router that requires configuration of redistribution. Selecting a single router for redistribution minimizes the likelihood of creating routing loops that are caused by feedback.
2. Determine which routing protocol is the core or backbone protocol. Typically, this protocol is OSPF, Intermediate System-to-Intermediate System Protocol (IS-IS), or EIGRP.
3. Determine which routing protocol is the edge or short-term (in the case of migration) protocol. Determine whether all routes from the edge protocol need to be propagated into the core. Consider methods that reduce the number of routes.
4. Select a method for injecting the required edge protocol routes into the core. Simple redistribution using summaries at network boundaries minimizes the number of new entries in the routing table of the core routers.

When you have planned the edge-to-core redistribution, consider how to inject the core routing information into the edge protocol. Your choice depends on your network.

Redistributing Routes into RIP

This topic describes how to redistribute routes into RIP.



Configuring Redistribution into RIP

```
RtrA(config)# router rip
RtrA(config-router)# redistribute ospf ?

 <1-65535>  Process ID
RtrA(config-router)# redistribute ospf 1 ?

  match      Redistribution of OSPF routes
  metric     Metric for redistributed routes
  route-map  Route map reference
  …
  <cr>
```

Default metric is infinity.

BSCI v3.0—5-3

Use this command to redistribute routes into RIP:

```
Router(config-router)# redistribute protocol [process-id]
[match route-type] [metric metric-value] [route-map map-tag]
```

Example: Configuring Redistribution into RIP

This figure shows how to configure for redistribution from OSPF process 1 into RIP.

In the figure, the example uses the **router rip** command to access the routing process into which routes need to be redistributed. In this case, it is the RIP routing process.

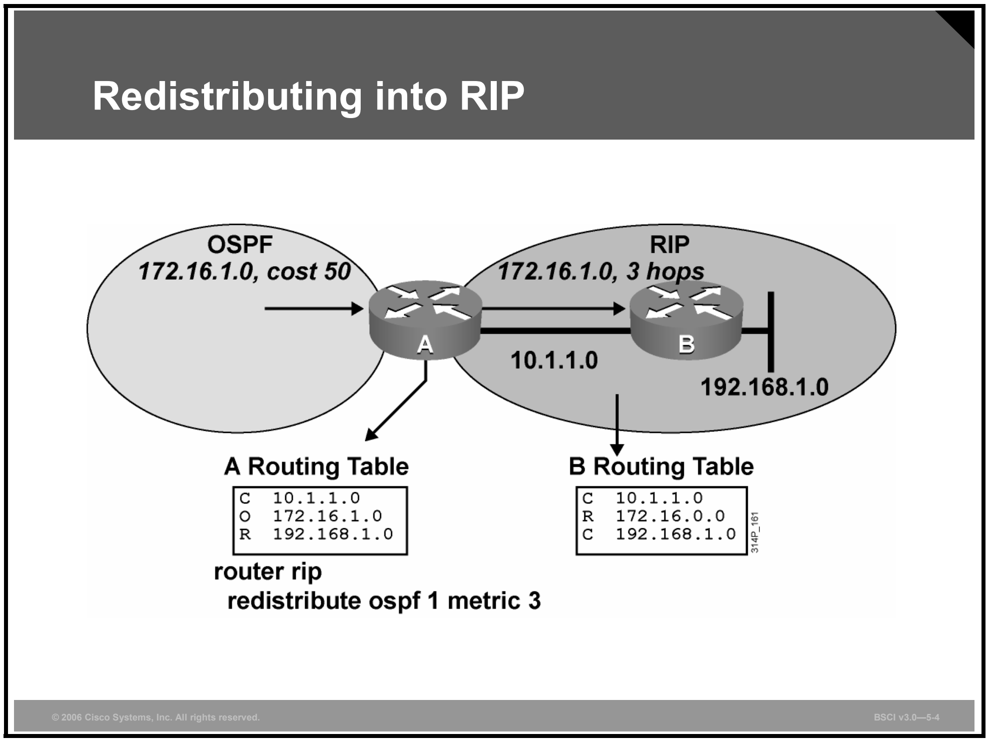
The example uses the **redistribute** command to specify the routing protocol to be redistributed into RIP. In this case, it is the OSPF routing process number 1.

Note The default metric is infinity except when you are redistributing a static or connected route. In that case, the default metric is 1.

The redistribute Command Parameters

This table details the parameters of the **redistribute** command.

Parameter	Description
protocol	Source protocol from which routes are being redistributed. It can be one of the following keywords: **connected**, **bgp**, **eigrp**, **egp**, **igrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **static**, or **rip**.
process-id	This value is an AS number, used for Border Gateway Protocol (BGP), Exterior Gateway Protocol (EGP), EIGRP, or IGRP. For OSPF, this value is an OSPF process ID.
match *route-type*	(Optional) Command parameter used for redistributing OSPF routes into another routing protocol. For OSPF, the criterion by which OSPF routes are redistributed into other routing domains. It can be any of the following: ■ **internal:** Redistributes routes that are internal to a specific AS. ■ **external 1:** Redistributes routes that are external to the AS, but are imported into OSPF as a type 1 external route. ■ **external 2:** Redistributes routes that are external to the AS, but are imported into OSPF as a type 2 external route.
metric *metric-value*	(Optional) Parameter used to specify the RIP seed metric for the redistributed route. When you are redistributing into RIP, this value is not specified and no value is specified using the **default-metric** router configuration command, then the default metric is 0, which is interpreted as infinity, and routes will not be redistributed. The metric for RIP is the hop count.
route-map *map-tag*	(Optional) Identifier of a configured route map to be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol.



Redistributing into RIP

A Routing Table

```
C  10.1.1.0
O  172.16.1.0
R  192.168.1.0
```

B Routing Table

```
C  10.1.1.0
R  172.16.0.0
C  192.168.1.0
```

```
router rip
 redistribute ospf 1 metric 3
```

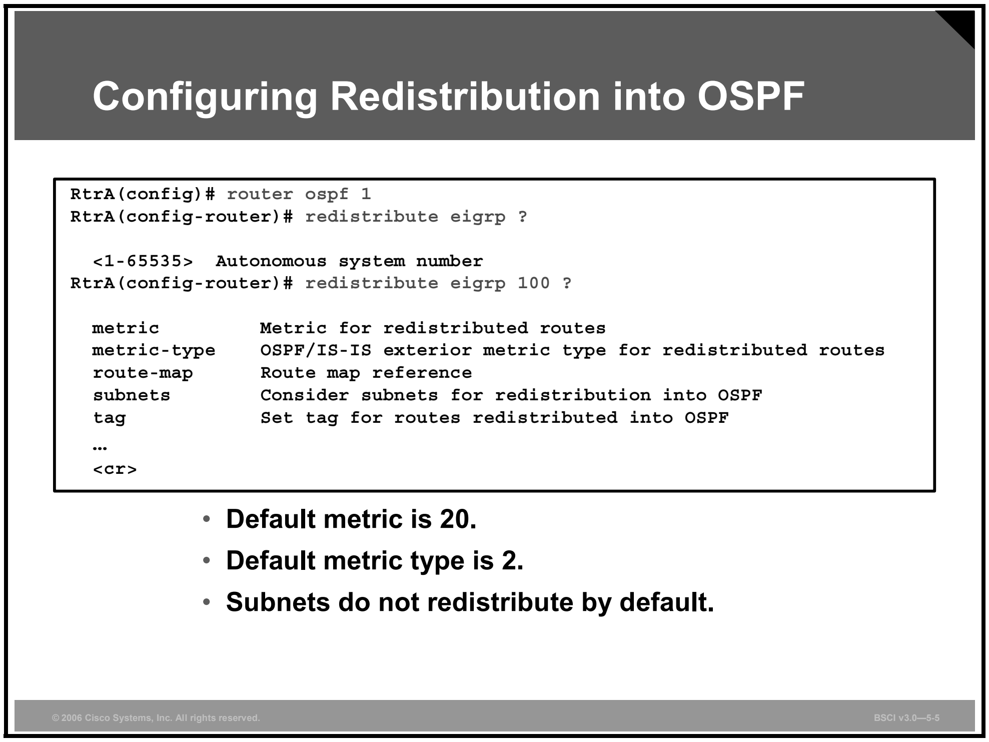
© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-4

Example: Redistributing into RIP

In the figure, routes from OSPF process number 1 are being redistributed into RIP and given a seed metric of 3. Because no route type is specified, both internal and external OSPF routes are redistributed into RIP.

Redistributing Routes into OSPF

This topic describes how to redistribute routes into OSPF.



Configuring Redistribution into OSPF

```
RtrA(config)# router ospf 1
RtrA(config-router)# redistribute eigrp ?

  <1-65535>  Autonomous system number
RtrA(config-router)# redistribute eigrp 100 ?

  metric         Metric for redistributed routes
  metric-type    OSPF/IS-IS exterior metric type for redistributed routes
  route-map      Route map reference
  subnets        Consider subnets for redistribution into OSPF
  tag            Set tag for routes redistributed into OSPF
  …
  <cr>
```

- **Default metric is 20.**
- **Default metric type is 2.**
- **Subnets do not redistribute by default.**

Use this command to redistribute routes into OSPF:

```
Router(config-router)# redistribute protocol [process-id]
[metric metric-value] [metric-type type-value] [route-map
map-tag] [subnets] [tag tag-value]
```

Example: Configuring Redistribution into OSPF

The figure shows how to configure for redistribution from EIGRP AS 100 into OSPF. It uses the **router ospf 1** command to access the OSPF routing process into which routes need to be redistributed. In this case, it is OSPF routing process 1.

The figure uses the **redistribute** command to specify the routing protocol to be redistributed into OSPF. In this case, it is the EIGRP routing process for AS 100.

The redistribute Command Parameters

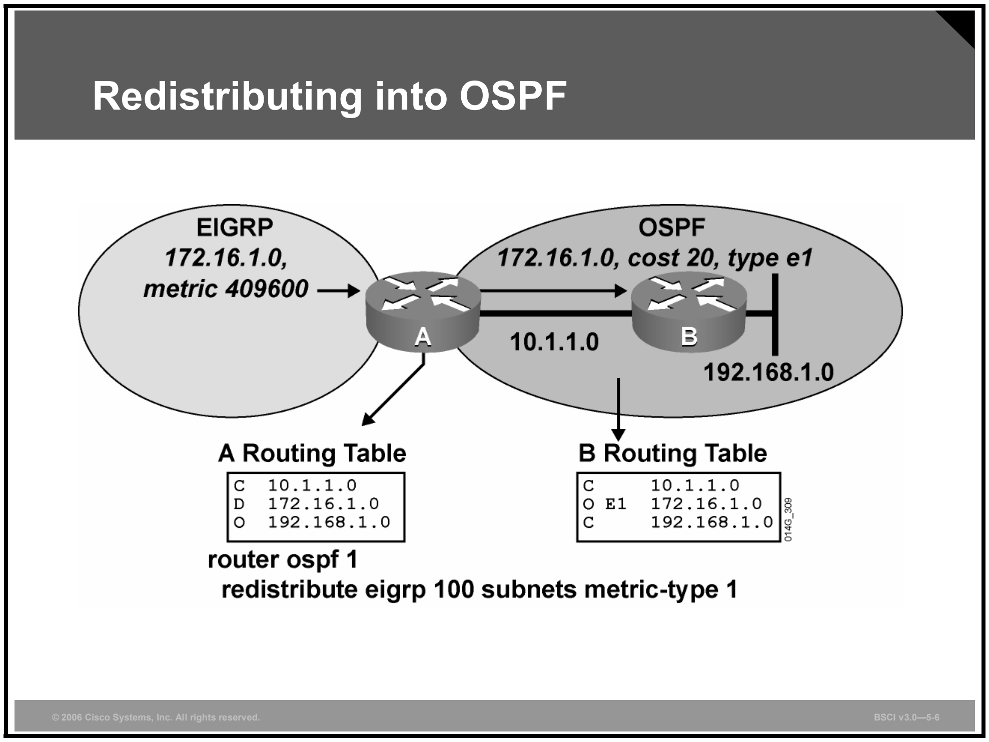
This table details more of the parameters of the **redistribute** command.

Parameter	Description
protocol	Source protocol from which routes are being redistributed. It can be one of the following keywords: **connected**, **bgp**, **eigrp**, **egp**, **igrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **static**, or **rip**.
process-id	This value is an AS number, used for BGP, EGP, EIGRP, or IGRP. For OSPF, this value is an OSPF process ID.
metric *metric-value*	(Optional) Parameter that specifies the OSPF seed metric that is used for the redistributed route. When you are redistributing into OSPF, the default metric is 20 (except for BGP, which is 1). Use a value consistent with the destination protocol, in this case, the OSPF cost.
metric-type *type-value*	(Optional) OSPF parameter that specifies the external link type that is associated with the external route that is advertised into the OSPF routing domain. This value can be 1 for type 1 external routes or 2 for type 2 external routes. The default is 2.
route-map *map-tag*	(Optional) Identifier of a configured route map to be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol.
subnets	(Optional) OSPF parameter that specifies that subnetted routes should be redistributed also. Only routes that are not subnetted are redistributed if the **subnets** keyword is not specified.
tag *tag-value*	(Optional) 32-bit decimal value that is attached to each external route. The OSPF protocol does not use this parameter. It may be used to communicate information between AS boundary routers (ASBRs).

Redistribution into OSPF can also be limited to a defined number of prefixes by the **redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only**] router configuration command. The threshold parameter will default to logging a warning at 75 percent of the defined maximum value configured.

After reaching the defined maximum number, no further routes are redistributed. If the **warning-only** parameter is configured, no limitation is placed on redistribution; the maximum value number simply becomes a second point where another warning messaged is logged.

This command was introduced in Cisco IOS Software Release 12.0(25)S and was integrated into Cisco IOS Software Release 12.2(18)S and 12.3(4)T and later.



Redistributing into OSPF

A Routing Table

```
C  10.1.1.0
D  172.16.1.0
O  192.168.1.0
```

B Routing Table

```
C     10.1.1.0
O E1  172.16.1.0
C     192.168.1.0
```

```
router ospf 1
 redistribute eigrp 100 subnets metric-type 1
```

BSCI v3.0—5-6

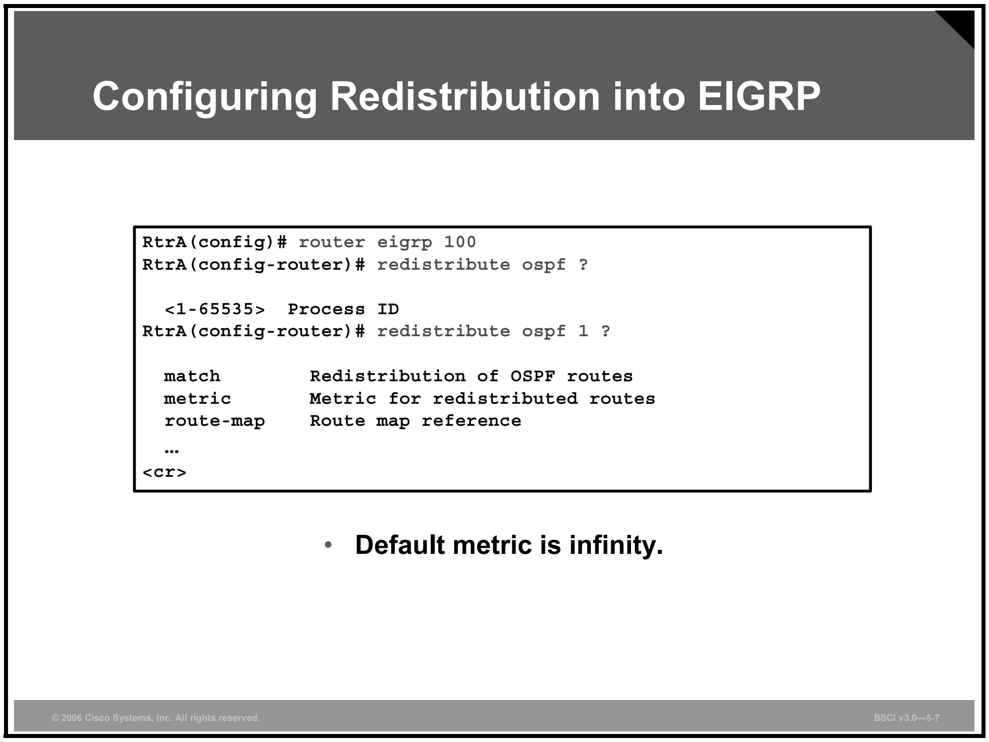
Example: Redistributing into OSPF

In this figure, the default metric of 20 for OSPF is being used, and the metric type is set to *external 1*. This setting means that the metric increases in increments whenever updates are passed through the network.

The command contains the **subnets** option, so subnets are redistributed.

Redistributing Routes into EIGRP

This topic describes how to redistribute routes into EIGRP.



Configuring Redistribution into EIGRP

```
RtrA(config)# router eigrp 100
RtrA(config-router)# redistribute ospf ?

  <1-65535>  Process ID
RtrA(config-router)# redistribute ospf 1 ?

  match        Redistribution of OSPF routes
  metric       Metric for redistributed routes
  route-map    Route map reference
  …
<cr>
```

- **Default metric is infinity.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-7

Use this command to redistribute routes into EIGRP:

```
router(config-router)# redistribute protocol [process-id]
[match {internal | external 1 | external 2}] [metric
metric-value] [route-map map-tag]
```

Example: Configuring Redistribution into EIGRP

The figure shows how to configure for redistribution from OSPF into EIGRP AS 100. It uses the **router eigrp 100** command to access the routing process into which routes need to be redistributed. In this case, it is the EIGRP routing process for AS 100.

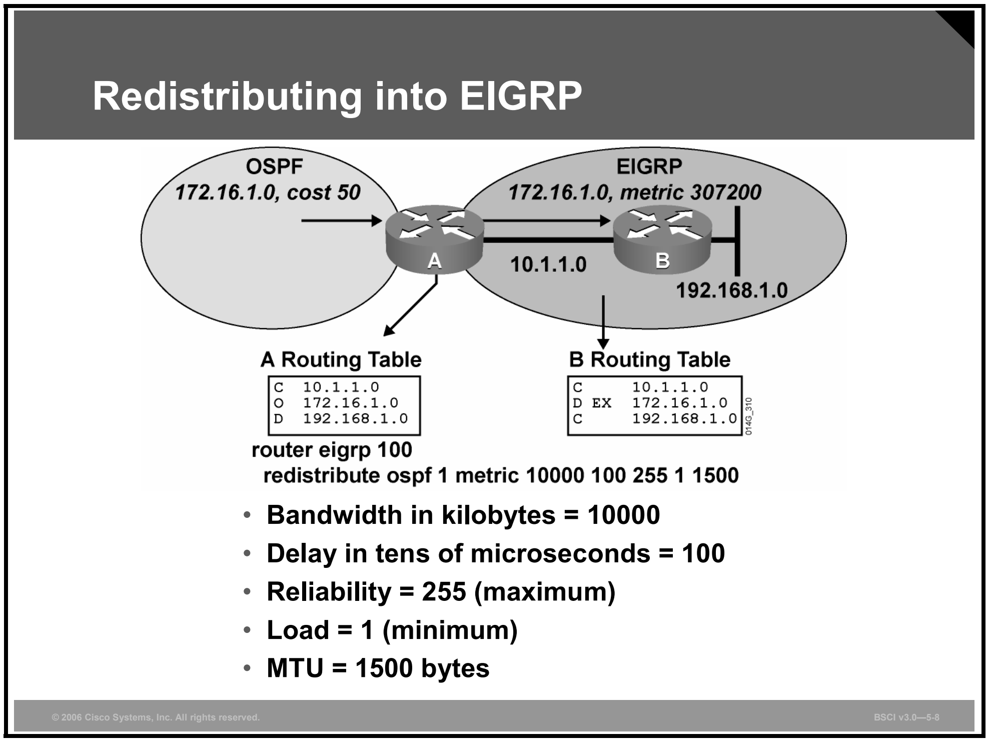
The figure uses the **redistribute** command to specify the routing protocol to be redistributed into EIGRP AS 100. In this case, it is OSPF routing process 1.

Note	When you are redistributing a static or connected route into EIGRP, the default metric is equal to the metric of the associated interface.

The redistribute Command Parameters

This table details the parameters of the **redistribute** command.

Parameter	Description
protocol	Source protocol from which routes are being redistributed. It can be one of the following keywords: **connected**, **bgp**, **eigrp**, **egp**, **igrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **static**, or **rip**.
process-id	This value is an AS number, used for BGP, EGP, EIGRP, or IGRP. For OSPF, this value is an OSPF process ID.
match *route-type*	(Optional) For OSPF, the criterion by which OSPF routes are redistributed into other routing domains. It can be one of the following: ■ **internal:** Redistributes routes that are internal to a specific AS. ■ **external 1:** Redistributes routes that are external to the AS but are imported into OSPF as a type 1 external route. ■ **external 2:** Redistributes routes that are external to the AS but are imported into OSPF as a type 2 external route.
metric *metric-value*	(Optional) Parameter that specifies the EIGRP seed metric, in the order of bandwidth, delay, reliability, load, and maximum transmission unit (MTU), for the redistributed route. When you are redistributing into protocols other than OSPF (including EIGRP), if this value is not specified and no value is specified using the **default-metric** router configuration command, the default metric is 0, zero is interpreted as infinity, and routes are not redistributed. Use a value consistent with the destination protocol. The metric for EIGRP is calculated based only on bandwidth and delay by default.
route-map *map-tag*	(Optional) Identifier of a configured route map that is interrogated to filter the importation of routes from this source routing protocol to the current routing protocol.



Redistributing into EIGRP

A Routing Table

```
C  10.1.1.0
O  172.16.1.0
D  192.168.1.0
```

B Routing Table

```
C     10.1.1.0
D EX  172.16.1.0
C     192.168.1.0
```

```
router eigrp 100
 redistribute ospf 1 metric 10000 100 255 1 1500
```

- **Bandwidth in kilobytes = 10000**
- **Delay in tens of microseconds = 100**
- **Reliability = 255 (maximum)**
- **Load = 1 (minimum)**
- **MTU = 1500 bytes**

BSCI v3.0—5-8

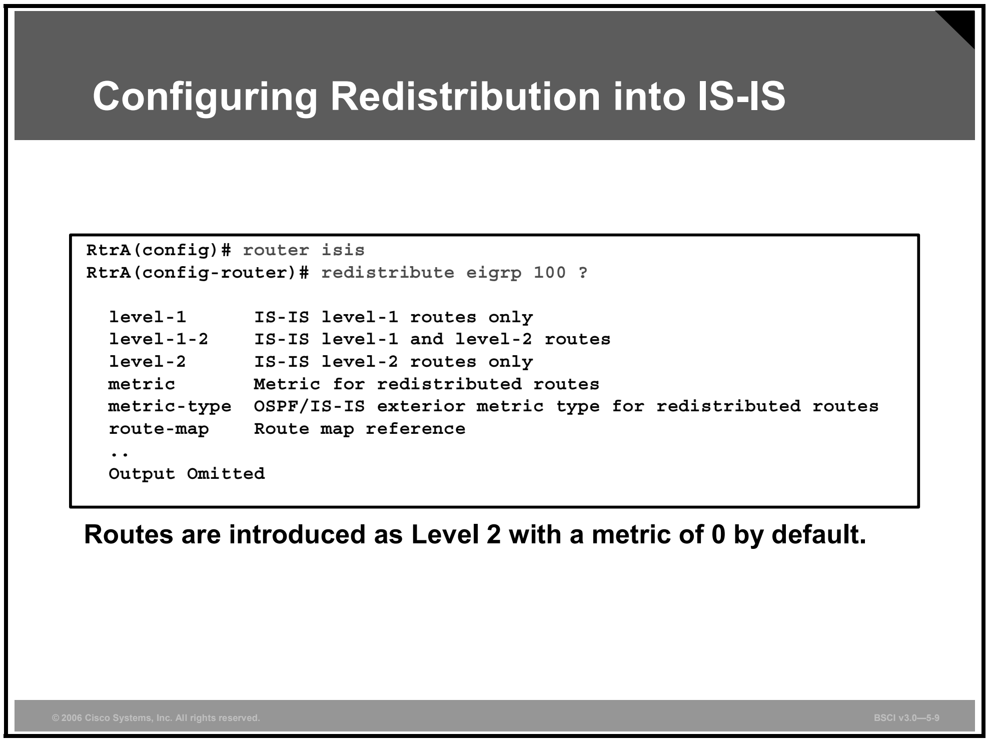
Example: Redistributing into EIGRP

In this figure, routes from OSPF process number 1 are redistributed into EIGRP AS 100. In this case, a metric is specified to ensure that routes are redistributed. The redistributed routes appear in the table of router B as external EIGRP (D EX) routes.

External EIGRP routes have a higher administrative distance than internal EIGRP (D) routes, so internal EIGRP routes are preferred over external EIGRP routes.

Redistributing Routes into IS-IS

This topic describes how to redistribute routes into IS-IS.



Configuring Redistribution into IS-IS

```
RtrA(config)# router isis
RtrA(config-router)# redistribute eigrp 100 ?

  level-1      IS-IS level-1 routes only
  level-1-2    IS-IS level-1 and level-2 routes
  level-2      IS-IS level-2 routes only
  metric       Metric for redistributed routes
  metric-type  OSPF/IS-IS exterior metric type for redistributed routes
  route-map    Route map reference
  ..
  Output Omitted
```

Routes are introduced as Level 2 with a metric of 0 by default.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-9

Use this command to redistribute routes into IS-IS:

```
router(config-router)# redistribute protocol [process-id]
[level level-value] [metric metric-value] [metric-type
type-value] [route-map map-tag]
```

Example: Configuring Redistribution into IS-IS

The figure shows how to configure for redistribution from EIGRP AS 100 into IS-IS. It uses the **router isis** command to access the routing process into which routes need to be redistributed. In this case, it is the IS-IS routing process.

The figure uses the **redistribute** command to specify the routing protocol to be redistributed into IS-IS. In this case, it is the EIGRP routing process for AS 100.

The redistribute Command Parameters

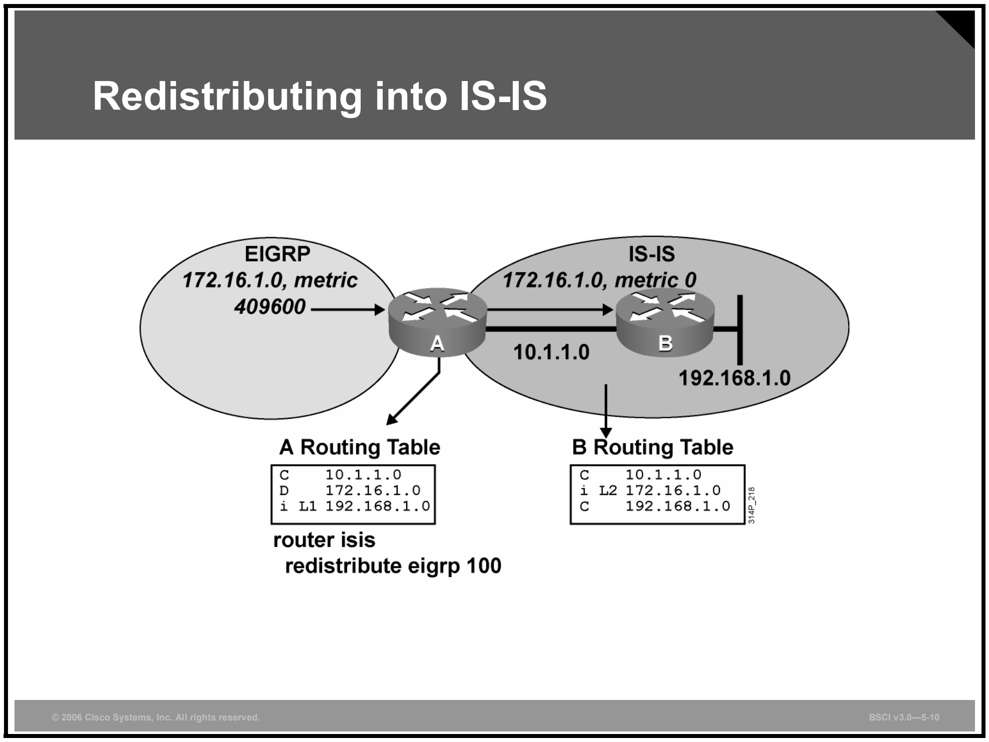
This table details the parameters of the **redistribute** command.

Parameter	Description
protocol	Specifies the source protocol from which routes are being redistributed. It can be one of the following keywords: **connected**, **bgp**, **eigrp**, **egp**, **igrp**, **isis**, **iso-igrp**, **mobile**, **odr**, **ospf**, **static**, or **rip**.
process-id	Specifies an AS number, used for BGP, EGP, EIGRP, or IGRP. For OSPF, this value is an OSPF process ID.
level *level-value*	Redistributes external routes as Level 1 (*level-1*), Level 1 and Level 2 (*level-1-2*), or Level 2 (*level-2*) routes. The default is Level 2.
metric *metric-value*	Specifies the IS-IS seed metric that is used for the redistributed route. IS-IS uses a default metric of 0. Unlike RIP, IGRP, and EIGRP, a default metric of 0 is not treated as unreachable and is redistributed. The metric is increased in increments as the route is propagated into the IS-IS domain. Use a value consistent with the destination protocol, in this case, the IS-IS cost.
metric-type *type-value*	Specifies the IS-IS metric type as external or internal. The default is internal.
route-map *map-tag*	(Optional) Specifies the identifier of a configured route map to be interrogated to filter the importation of routes from this source routing protocol to the current routing protocol.

When redistributing IS-IS routes into other routing protocols, you have the option to include Level 1, Level 2, or both Level 1 and Level 2 routes. The output shows the parameters available for choosing these routes; if no level is specified, then all routes are redistributed.

```
Router(config)# router ospf 1
Router(config-router)# redistribute isis ?
  <output omitted>
  level-1              IS-IS level-1 routes only
  level-1-2            IS-IS level-1 and level-2 routes
  level-2              IS-IS level-2 routes only
  <output omitted>
```

Redistribution into IS-IS can also be limited to a defined number of prefixes by the **redistribute maximum-prefix** *maximum* [*threshold*] [**warning-only** | **withdraw**] router configuration command. The threshold parameter will default to logging a warning at 75 percent of the defined maximum value configured. After reaching the defined maximum number, no further routes are redistributed. The optional **withdraw** parameter will also cause IS-IS to rebuild link-state protocol data units (PDUs) (link-state packets [LSPs]) without the external (redistributed) IP prefixes. If the **warning-only** parameter is configured, no limitation is placed on redistribution. The maximum value number simply becomes a second point where another warning messaged is logged. This command was introduced in Cisco IOS Software Release 12.0(25)S and was integrated into Cisco IOS Software Releases 12.2(18)S and 12.3(4)T and later.



Redistributing into IS-IS

A Routing Table

```
C    10.1.1.0
D    172.16.1.0
i L1 192.168.1.0
```

B Routing Table

```
C    10.1.1.0
i L2 172.16.1.0
C    192.168.1.0
```

```
router isis
 redistribute eigrp 100
```

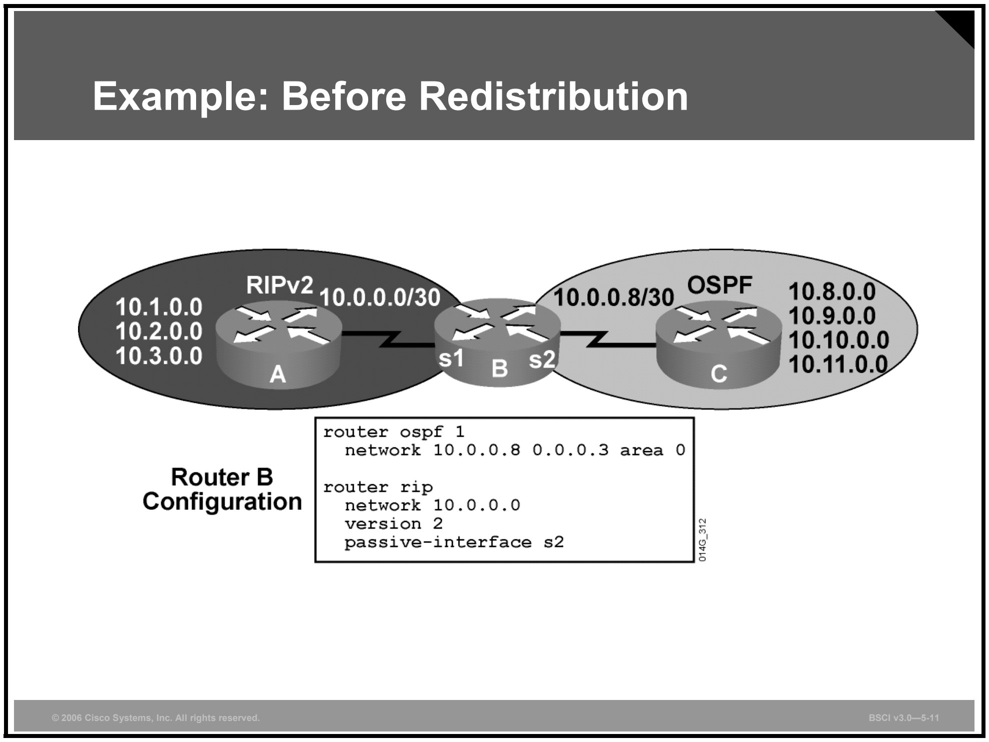
Example: Redistributing into IS-IS

In this figure, routes are redistributed from EIGRP AS 100 into IS-IS on router A. No metric is given, so these routes have a seed metric of 0.

No level type is given, so the routes are redistributed as Level 2 routes (as displayed in the router B routing table).

Verifying Route Redistribution

This topic describes how to verify route redistribution operations.



Example: Before Redistribution

Router B Configuration

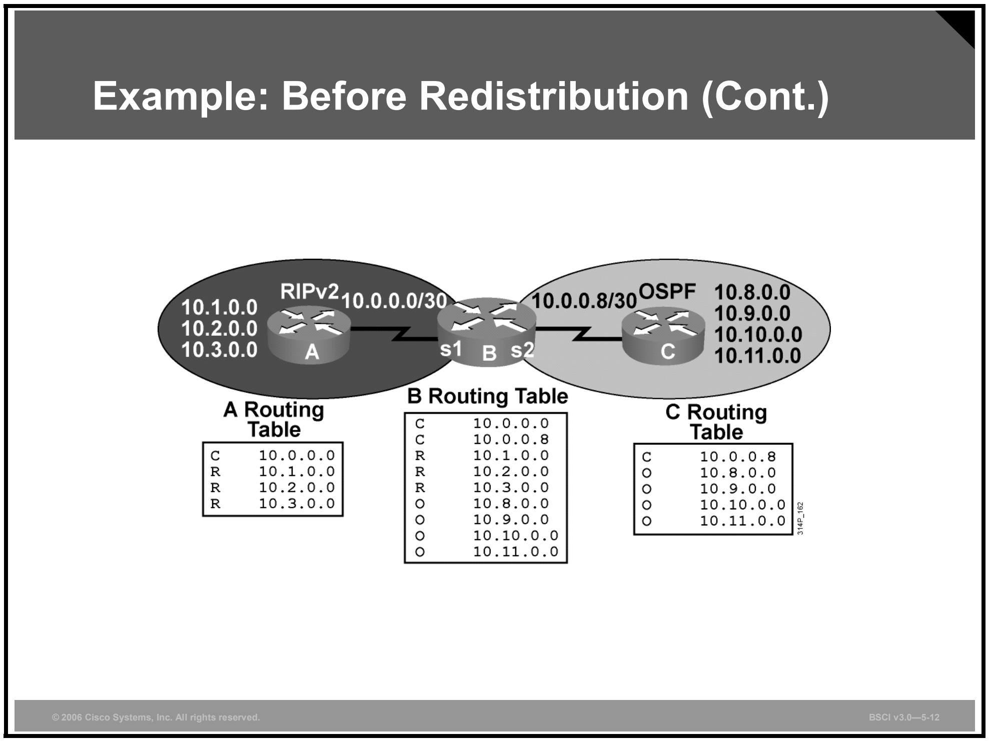
```
router ospf 1
  network 10.0.0.8 0.0.0.3 area 0

router rip
  network 10.0.0.0
  version 2
  passive-interface s2
```

Example: Before Redistribution

This figure shows the network of a hypothetical company. The network begins with two routing domains, or autonomous systems, one using OSPF and one using RIP version 2 (RIPv2). Router B is the boundary router. Router B connects directly to one router within each routing domain and runs both protocols.

Router A is in the RIP domain, and is advertising subnets 10.1.0.0, 10.2.0.0, and 10.3.0.0 to router B. Router C is in the OSPF domain and is advertising subnets 10.8.0.0, 10.9.0.0, 10.10.0.0, and 10.11.0.0 to router B.

The configuration of router B is shown in the figure. RIP is required to run on the serial 1 interface only; therefore, the **passive-interface** command is given for interface serial 2. The **passive-interface** command prevents RIP from sending route advertisements out that interface. OSPF is configured on serial 2.



Example: Before Redistribution (Cont.)

A Routing Table

C	10.0.0.0
R	10.1.0.0
R	10.2.0.0
R	10.3.0.0

B Routing Table

C	10.0.0.0
C	10.0.0.8
R	10.1.0.0
R	10.2.0.0
R	10.3.0.0
O	10.8.0.0
O	10.9.0.0
O	10.10.0.0
O	10.11.0.0

C Routing Table

C	10.0.0.8
O	10.8.0.0
O	10.9.0.0
O	10.10.0.0
O	10.11.0.0

BSCI v3.0—5-12

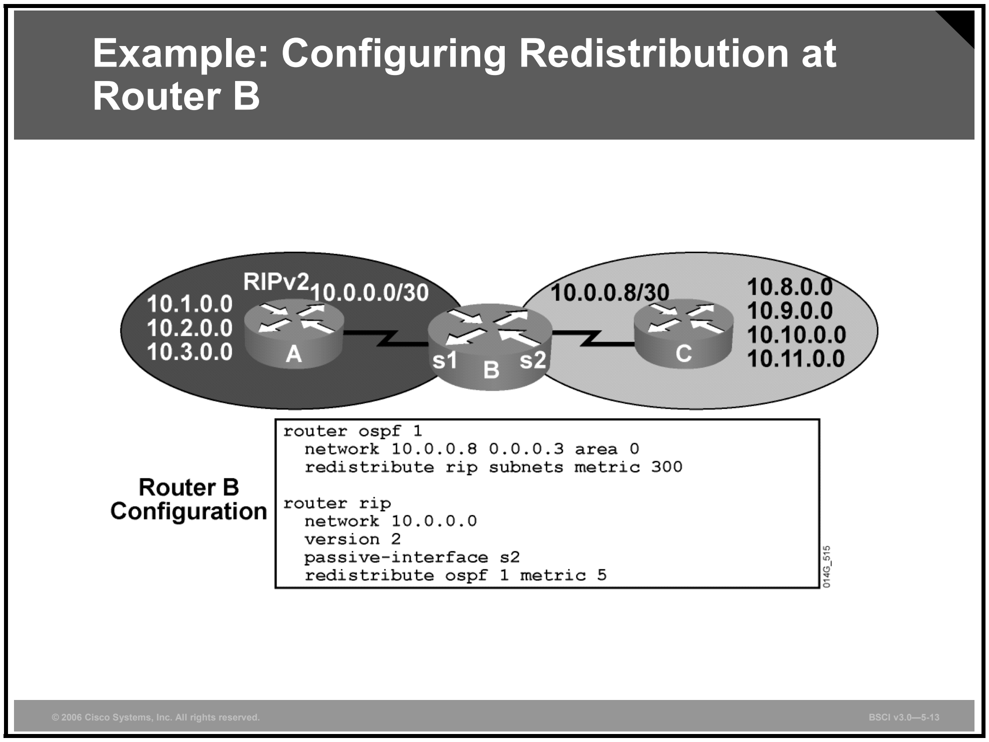
Example: Routing Tables Before Redistribution

This figure shows the routing tables of routers A, B, and C. Each routing domain is separate, and routers within them recognize routes that are communicated from their own routing protocols only.

The only router with information on all the routes is router B, which is the boundary router that runs both routing protocols and connects to both routing domains.

The goal of redistribution in this network is for all routers to recognize all routes within the company. To accomplish this goal, redistribution is planned:

- Redistribute RIP routes into OSPF.
- Redistribute OSPF routes into the RIP domain.



Example: Configuring Redistribution at Router B

Router B Configuration

```
router ospf 1
  network 10.0.0.8 0.0.0.3 area 0
  redistribute rip subnets metric 300

router rip
  network 10.0.0.0
  version 2
  passive-interface s2
  redistribute ospf 1 metric 5
```

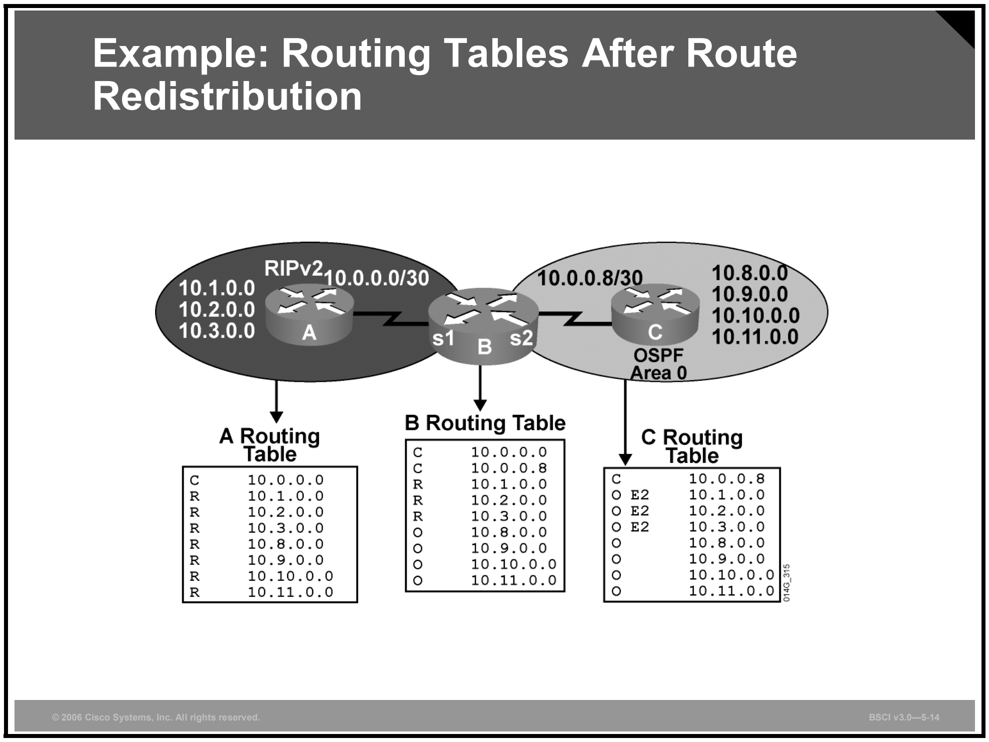
Example: Configuring Redistribution

Router B is the boundary router, so redistribution is configured on it. This figure shows how router B is configured to accomplish the required redistribution.

RIP is redistributed under the OSPF process. In this example, the metric is set under the **redistribute** command. Other options include specifying a default metric or accepting the OSPF default metric of 20.

The **default-metric** command assigns a seed metric to all routes redistributed into OSPF from any origin. If a metric value is configured under a specific **redistribute** command, this value overrides the default metric value. A value of 300 is selected because it is a worse metric than any of the native OSPF routes.

Under the RIP process, routes are redistributed in from OSPF process number 1. These routes are redistributed into RIP with a metric of 5. A value of 5 is chosen because it is higher than any metric in the RIP network.



Example: Routing Tables After Route Redistribution

A Routing Table

```
C      10.0.0.0
R      10.1.0.0
R      10.2.0.0
R      10.3.0.0
R      10.8.0.0
R      10.9.0.0
R      10.10.0.0
R      10.11.0.0
```

B Routing Table

```
C      10.0.0.0
C      10.0.0.8
R      10.1.0.0
R      10.2.0.0
R      10.3.0.0
O      10.8.0.0
O      10.9.0.0
O      10.10.0.0
O      10.11.0.0
```

C Routing Table

```
C          10.0.0.8
O E2       10.1.0.0
O E2       10.2.0.0
O E2       10.3.0.0
O          10.8.0.0
O          10.9.0.0
O          10.10.0.0
O          10.11.0.0
```

Example: Routing Tables After Route Redistribution

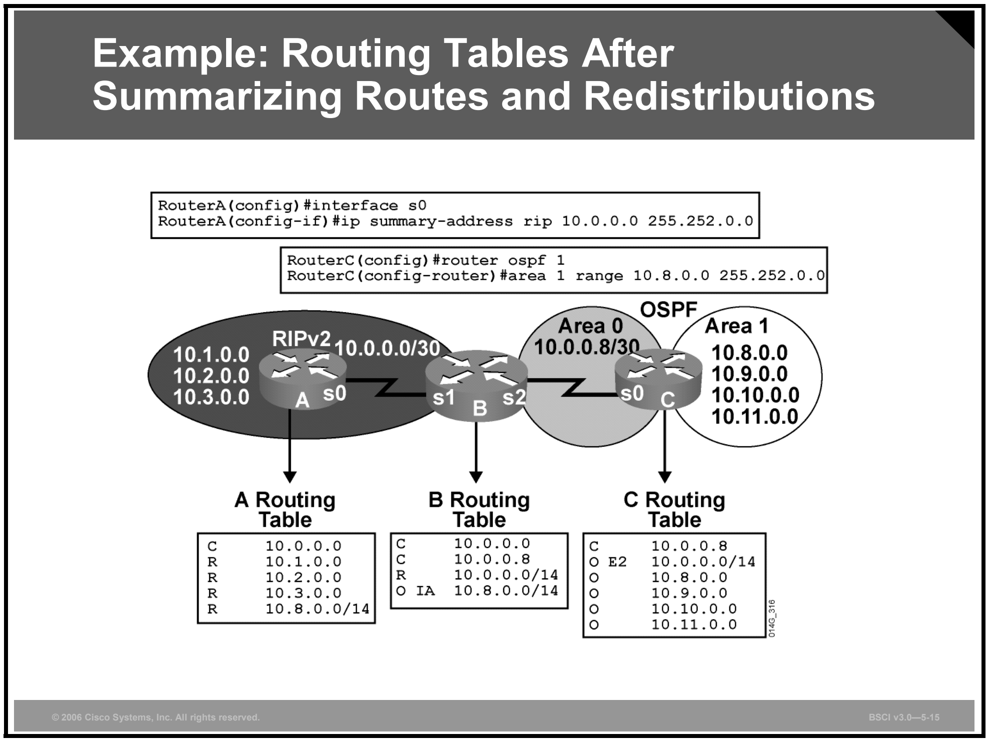
This figure shows the routing tables of all three routers after redistribution is completed. The goal is accomplished. All routers now have routes to all remote subnets. There is complete reachability within the entire network.

Routers A and C now have many more routes to keep track of than before. Each router is also affected by topology changes in the routing domain of the other router.

Depending on network requirements, you can increase efficiency by summarizing the routes before redistributing them. Remember that route summarization hides information.

If routers in the other autonomous systems are required to track topology changes within the network, then route summarization should not be performed, because it hides information that the routers need.

A more typical case is that the routers need to recognize topology changes only within their own routing domains. In this case, performing route summarization is appropriate.



Example: Routing Tables After Summarizing Routes and Redistributions

```
RouterA(config)#interface s0
RouterA(config-if)#ip summary-address rip 10.0.0.0 255.252.0.0
```

```
RouterC(config)#router ospf 1
RouterC(config-router)#area 1 range 10.8.0.0 255.252.0.0
```

A Routing Table

```
C      10.0.0.0
R      10.1.0.0
R      10.2.0.0
R      10.3.0.0
R      10.8.0.0/14
```

B Routing Table

```
C      10.0.0.0
C      10.0.0.8
R      10.0.0.0/14
O IA   10.8.0.0/14
```

C Routing Table

```
C      10.0.0.8
O E2   10.0.0.0/14
O      10.8.0.0
O      10.9.0.0
O      10.10.0.0
O      10.11.0.0
```

If routes are summarized before redistribution, then the routing tables of each router are significantly smaller. Router B benefits the most; it now has only four routes to keep track of instead of nine. Router A has five routes instead of eight, and router C has six routes to keep track of instead of eight.

These commands are used to summarize routes for each protocol:

- **Router A, RIP:** For RIPv2, the summarization command is given at the interface connecting router B with router A. This summary address is advertised out of that interface instead of the individual subnets. One limitation of RIP is that the subnet mask of the summary address must be greater than or equal to the default mask for the major classful network. Use this summarization command for RIPv2:

```
RouterA(config)# interface s0
RouterA(config-if)# ip summary-address rip 10.0.0.0
255.252.0.0
```

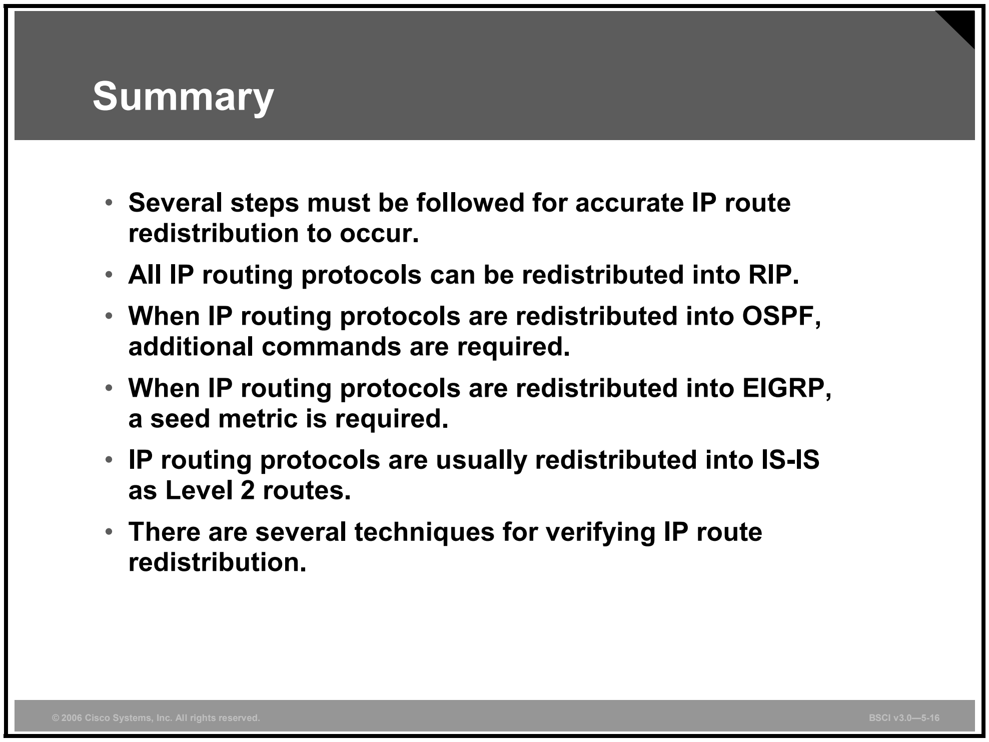
Note This summary includes 10.0.0.0, which is acceptable in this case because it is directly connected with a longer mask.

- **Router C, OSPF:** You must perform summarization in OSPF at an area border router (ABR) or an ASBR. Create another OSPF area that includes the four subnets to be summarized. Give the command for summarization under the OSPF process at router C, which becomes an ABR. Use this summarization command for OSPF:

```
RouterC(config)# router ospf 1
RouterC(config-router)# area 1 range 10.8.0.0 255.252.0.0
```

Summary

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



Summary

- Several steps must be followed for accurate IP route redistribution to occur.
- All IP routing protocols can be redistributed into RIP.
- When IP routing protocols are redistributed into OSPF, additional commands are required.
- When IP routing protocols are redistributed into EIGRP, a seed metric is required.
- IP routing protocols are usually redistributed into IS-IS as Level 2 routes.
- There are several techniques for verifying IP route redistribution.

Lesson 3

Controlling Routing Update Traffic

Overview

Routing updates compete with user data for bandwidth and router resources, yet routing updates are critical because they carry the information that routers need to make sound routing decisions.

To ensure that the network operates efficiently, you must control and tune routing updates. Information about networks must be sent where it is needed and filtered from where it is not needed.

There is no one type of route filter that is appropriate for every situation. Therefore, the more techniques that you have at your disposal, the better your chance of having a smooth, well-run network.

This lesson discusses how to control the updates that are sent and received by dynamic routing protocols and how to control the routes that are redistributed into routing protocols. You can control routing updates by using the **passive-interface** command, deploying route maps, and manipulating administrative distance.

Objectives

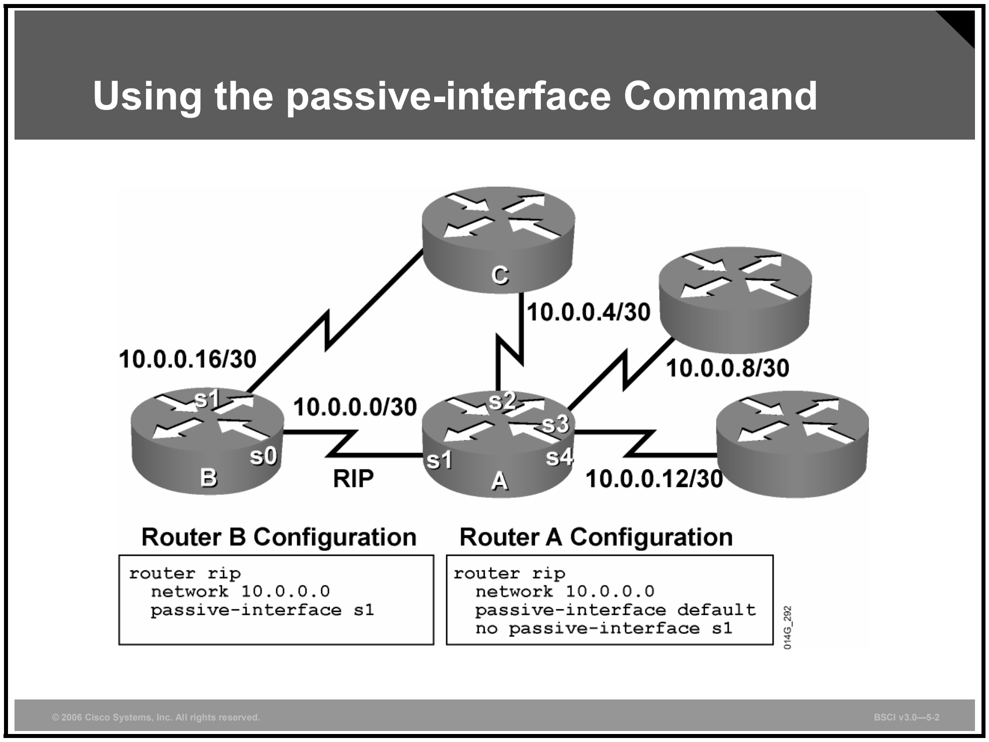
Upon completing this lesson, you will be able to configure dynamic routing protocol updates for passive interfaces and distribute lists. This ability includes being able to meet these objectives:

- Describe how to configure a passive interface
- Describe how to configure route filtering using distribute lists
- Explain how to implement the distribute list route-filtering technique
- Describe the functionality of route maps

- Describe how to use the route-map command to define the conditions for route filtering and redistribution
- Describe how to implement route maps with route redistribution
- Describe the features of administrative distance in terms of routing protocols
- Describe how to modify administrative distance on the router globally for a particular routing protocol or specifically for certain routes
- Describe the impact of administrative distance changes on routing tables

Configuring a Passive Interface

This topic describes how to configure a passive interface.



Using the passive-interface Command

Router B Configuration

```
router rip
  network 10.0.0.0
  passive-interface s1
```

Router A Configuration

```
router rip
  network 10.0.0.0
  passive-interface default
  no passive-interface s1
```

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-2

There are times when you must include an interface in a **network** command, although you do not want that interface to participate in the routing protocol. The **passive-interface** command prevents routing updates for a routing protocol from being sent through a router interface. The **passive-interface** command can set a particular interface or all router interfaces to passive. Use the **default** option to set all router interfaces.

With Internet service providers (ISPs) and large enterprise networks, many of the distribution routers have more than 200 interfaces. Before the introduction of the passive interface default feature in Cisco IOS Software Release 12.0, the solution to the numerous interface problems was to configure the routing protocol on all interfaces and manually set the **passive-interface** command on the interfaces where you did not require adjacency.

This solution meant entering 200 or more passive-interface statements. You can now solve this configuration scalability problem by using a single **passive-interface default** command to set all interfaces to passive by default. You then enable routing on individual interfaces where you require adjacencies using the **no passive-interface** command.

When you use the **passive-interface** command with Routing Information Protocol (RIP) and Interior Gateway Routing Protocol (IGRP), routing updates are not sent to the specified interface. The router still receives routing updates from that interface.

When you use the **passive-interface** command with Enhanced Interior Gateway Routing Protocol (EIGRP), hello messages are not sent to the specified interface. Neighboring router relationships do not form with other routers that are reachable through that interface. Because no neighbors are found on an interface, no other EIGRP traffic is sent.

Using the **passive-interface** command on a router running a link-state routing protocol also prevents the router from establishing neighboring router adjacencies with other routers that are connected to the link that is specified in the command.

The router does not send hellos to the specified interface. Therefore, you cannot establish neighbor adjacencies, because the Hello protocol is used to verify bidirectional communication between routers.

To configure a passive interface, regardless of the routing protocol, use the following procedure:

Step 1 Select the router and routing protocol that require the passive interface.

Step 2 Determine the interfaces through which you do not want routing update traffic (or hellos for link-state routing protocols and EIGRP) to be sent.

Step 3 Configure the router using the **passive-interface** command.

The passive-interface Command Parameters

The table describes the parameters of the **passive-interface** command.

Parameter	Description
`type number` *`interface-number`*	Specifies type of interface and interface number that does not send routing updates or hellos for link-state routing protocols and EIGRP
`default`	Sets all interfaces on the router as passive by default

Example: Using the passive interface Command

In the previous figure, routers A and B run RIP, and have a network statement that encompasses all their interfaces; however, you want to run RIP on the link between router A and router B only.

Router A has several interfaces; the **passive-interface default** command was used to set all interfaces to passive, and then the **no passive-interface** command was used to enable the one interface from which RIP updates are desired. Router B has only two interfaces; the **passive-interface** command was used for the one interface that is not to participate in RIP routing.

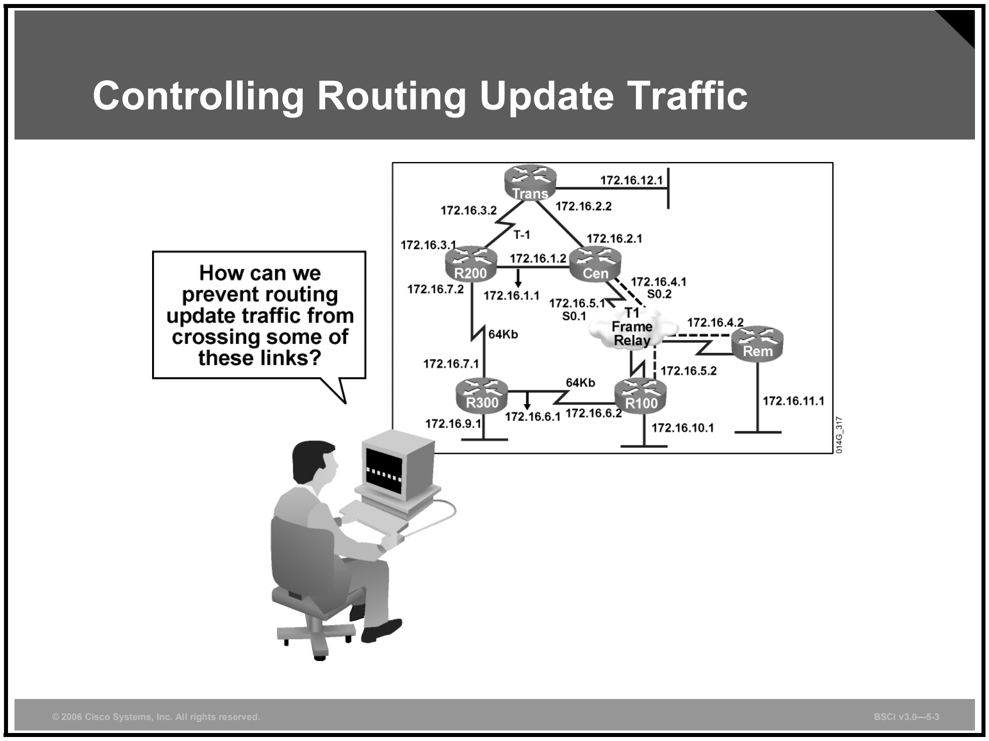
It is important to understand how this configuration affects the information that is exchanged between routers A and B and between them and router C. Unless you configure another routing protocol and redistribute between it and RIP, router A does not tell router C that it has a way to reach the networks advertised by router B via RIP.

Likewise, router B does not tell router C that it has a way to reach the networks advertised by router A via RIP.

Redundancy is built into this network. However, the three routers are not able to use the redundancy effectively. For example, if the link between router C and router A fails, router C does not know that it has an alternate route through router B.

Configuring Route Filtering Using Distribute Lists

This topic describes how to configure route filtering using distribute lists.



Controlling Routing Update Traffic

How can we prevent routing update traffic from crossing some of these links?

BSCI v3.0—5-3

The passive interface technique prevents all routing updates from being advertised out of an interface. However, in many cases you do not want to prevent all routing information from being advertised.

You might want to block the advertisement of only certain specific routes. For example, you could use such a solution to prevent routing loops when you are implementing two-way route redistribution with dual redistribution points.

Some ways to control or prevent dynamic routing updates are as follows:

- **Passive interface:** As previously stated, this feature prevents all routing updates from being sent through an interface. For EIGRP, Open Shortest Path First Protocol (OSPF), and Intermediate System-to-Intermediate System Protocol (IS-IS), this method includes Hello protocol packets.
- **Default routes:** This feature instructs the router that if it does not have a route for a given destination, it should send the packet to the default route. Therefore, no dynamic routing updates about the remote destinations are necessary.
- **Static routes:** This feature allows routes to remote destinations to be manually configured in the router. Therefore, no dynamic routing updates about the remote destinations are necessary.

Another way to control routing updates is a technique called a “distribute list.” A distribute list allows the application of an access control list (ACL) to routing updates. You may be familiar with ACLs associated with an interface and used to control IP traffic. However, routers can have many interfaces, and route information can also be obtained through route redistribution, which does not involve an interface at all.

Additionally, ACLs do not affect traffic that is originated by the router, so applying one to an interface would have no effect on outgoing routing advertisements. When you link an ACL to a distribute list, routing updates can be controlled no matter what their source is.

Configure ACLs in global configuration mode, and then configure the associated distribute list under the routing protocol. The ACL should permit the networks that will be advertised or redistributed and deny the networks that will remain hidden.

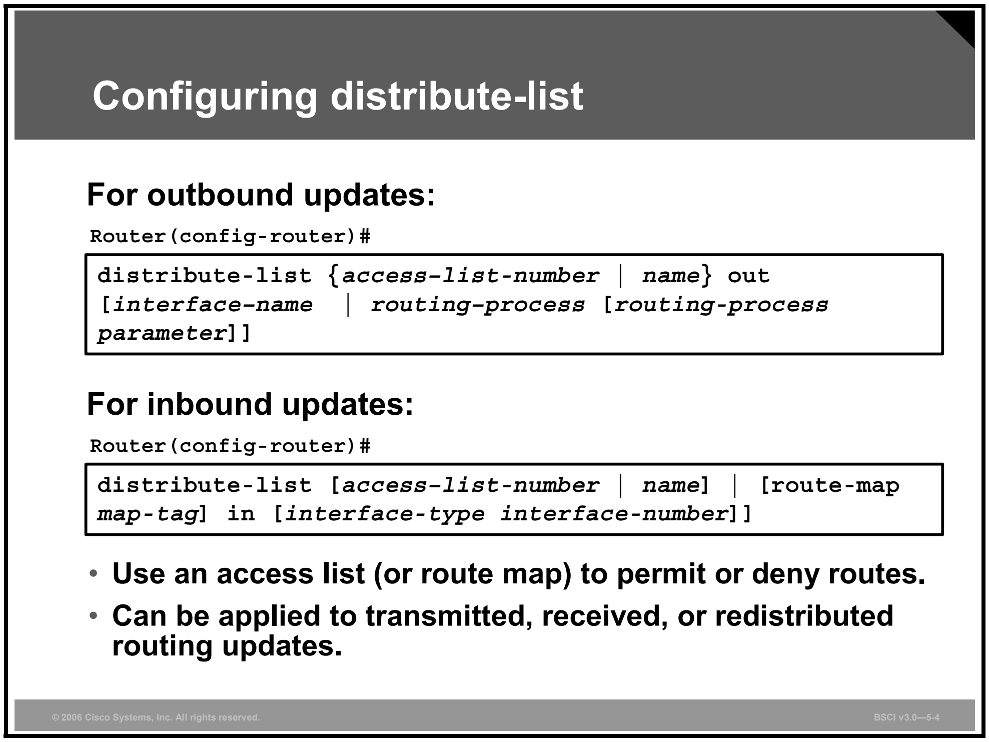
The router then applies the ACL to routing updates for that protocol. Options in the **distribute-list** command allow updates to be filtered based on three factors:

- Incoming interface
- Outgoing interface
- Redistribution from another routing protocol

Using a distribute list gives the administrator great flexibility in determining just which routes the router distributes.

Implementing the Distribute List

This topic describes how to implement the distribute list route-filtering technique.



Configuring distribute-list

For outbound updates:

```
Router(config-router)#
distribute-list {access-list-number | name} out
[interface-name  | routing-process [routing-process
parameter]]
```

For inbound updates:

```
Router(config-router)#
distribute-list [access-list-number | name] | [route-map
map-tag] in [interface-type interface-number]]
```

- **Use an access list (or route map) to permit or deny routes.**
- **Can be applied to transmitted, received, or redistributed routing updates.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-4

You can filter routing update traffic for any protocol by defining an ACL and applying it to a specific routing protocol. You use the **distribute-list** command and link it to an ACL to complete the filtering of routing update traffic. (The inbound **distribute-list** command allows the use of a route map instead of an ACL.)

A distribute list enables the filtering of routing updates coming into a specific interface from neighboring routers using the same routing protocol or going out of the interface toward the routers. A distribute list also allows the filtering of routes redistributed from other routing protocols or sources. To configure a distribute list using an ACL, use the following procedure:

1. Identify the network addresses that you want to filter and create an ACL.
2. Determine whether you want to filter traffic on an incoming interface, an outgoing interface, or routes being redistributed from another routing source.
3. Use the **distribute-list out** command to assign the ACL to filter outgoing routing updates or to assign it to routes being redistributed into the protocol. The **distribute-list out** command cannot be used with link-state routing protocols for blocking outbound link-state advertisements (LSAs) on an interface.
4. Use the **distribute-list in** command to assign the ACL to filter incoming routing updates coming in through an interface. This command prevents most routing protocols from placing the filtered routes in their database. When this command is used with OSPF, the routes are placed in the database but not the routing table.

The distribute-list out Command Parameters

This table describes the parameters of the **distribute-list out** command.

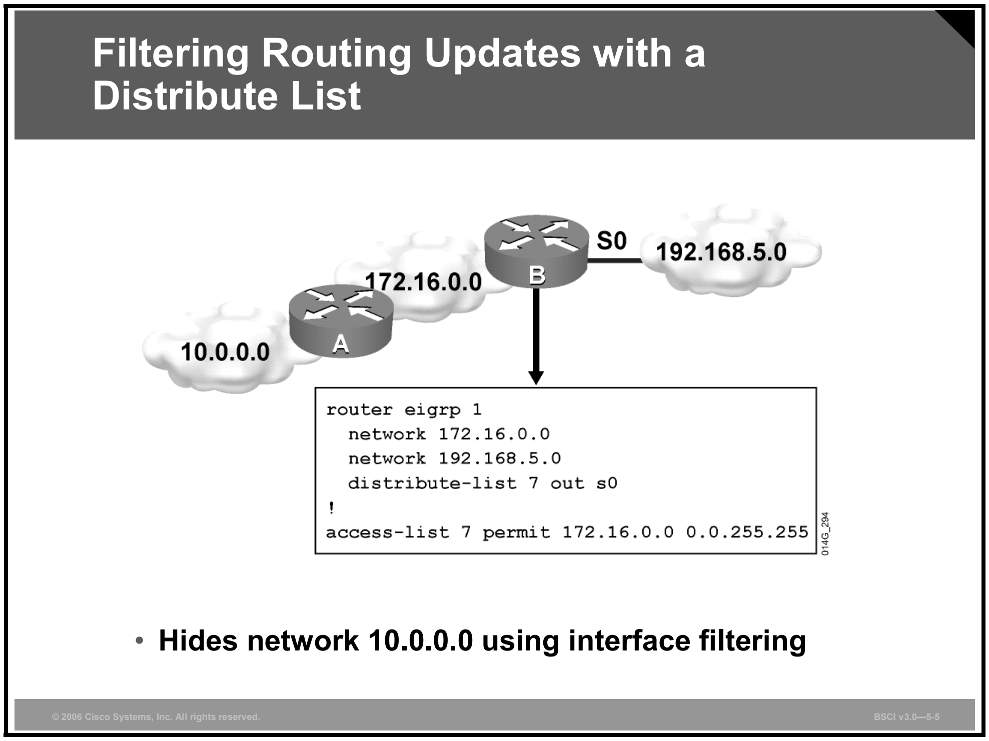
Parameter	Description
access-list-number | name	Specifies standard ACL number or name
out	Applies the ACL to outgoing routing updates
interface-name	(Optional) Specifies name of interface out of which updates are filtered
routing-process	(Optional) Specifies name of the routing process, or the keyword **static** or **connected**, that is being redistributed and from which updates are filtered
routing-process parameter	(Optional) Specifies routing process parameter, such as the autonomous system (AS) number of the routing process

To assign the ACL to filter incoming routing updates, use the **distribute-list in** command.

The distribute-list in Command Parameters

This table describes the parameters of the **distribute-list in** command.

Parameter	Description
access-list-number | name	Specifies standard ACL number or name.
map-tag	(Optional) Name of the route map that defines which networks are to be installed in the routing table and which are to be filtered from the routing table. This argument is supported by OSPF only.
in	Applies the ACL to incoming routing updates.
interface-type interface-number	(Optional) Specifies interface type and number from which updates are filtered.



Filtering Routing Updates with a Distribute List

```
router eigrp 1
  network 172.16.0.0
  network 192.168.5.0
  distribute-list 7 out s0
!
access-list 7 permit 172.16.0.0 0.0.255.255
```

- **Hides network 10.0.0.0 using interface filtering**

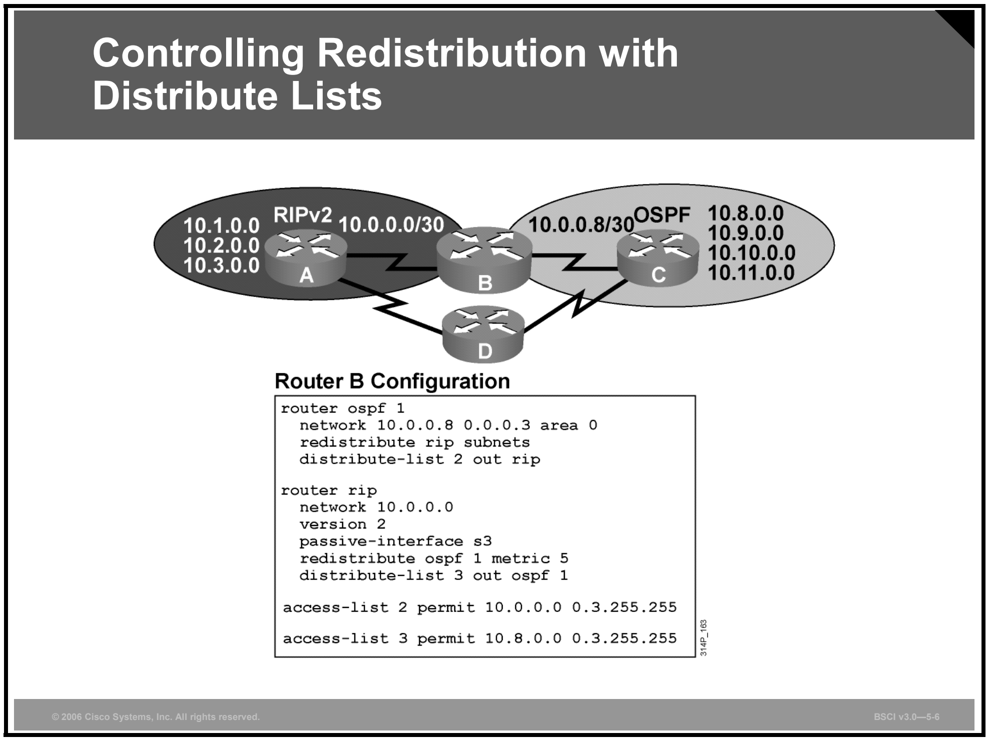
The distribute-list Configuration Commands

This table describes some of the commands shown in the figure.

Command	Description
`distribute-list 7 out s0`	Applies ACL 7 as a route filter on EIGRP routing updates sent out interface serial 0
`access-list 7 permit 172.16.0.0 0.0.255.255`	Configures a standard ACL to permit routing information regarding only the 172.16.0.0 network

The **distribute-list 7 out s0** command applies ACL 7 to routing updates sent out from interface serial 0 to other routers running this routing protocol. This ACL permits routing information about network 172.16.0.0 only.

The implicit deny any at the end of the ACL prevents routing updates about any other networks from being advertised. As a result, network 10.0.0.0 is hidden from the rest of the network.



Controlling Redistribution with Distribute Lists

Router B Configuration

```
router ospf 1
  network 10.0.0.8 0.0.0.3 area 0
  redistribute rip subnets
  distribute-list 2 out rip

router rip
  network 10.0.0.0
  version 2
  passive-interface s3
  redistribute ospf 1 metric 5
  distribute-list 3 out ospf 1

access-list 2 permit 10.0.0.0 0.3.255.255

access-list 3 permit 10.8.0.0 0.3.255.255
```

With mutual redistribution, using a distribute list helps prevent route feedback, which also helps prevent routing loops. Route feedback occurs when routes originally learned from one routing protocol are redistributed back into that protocol.

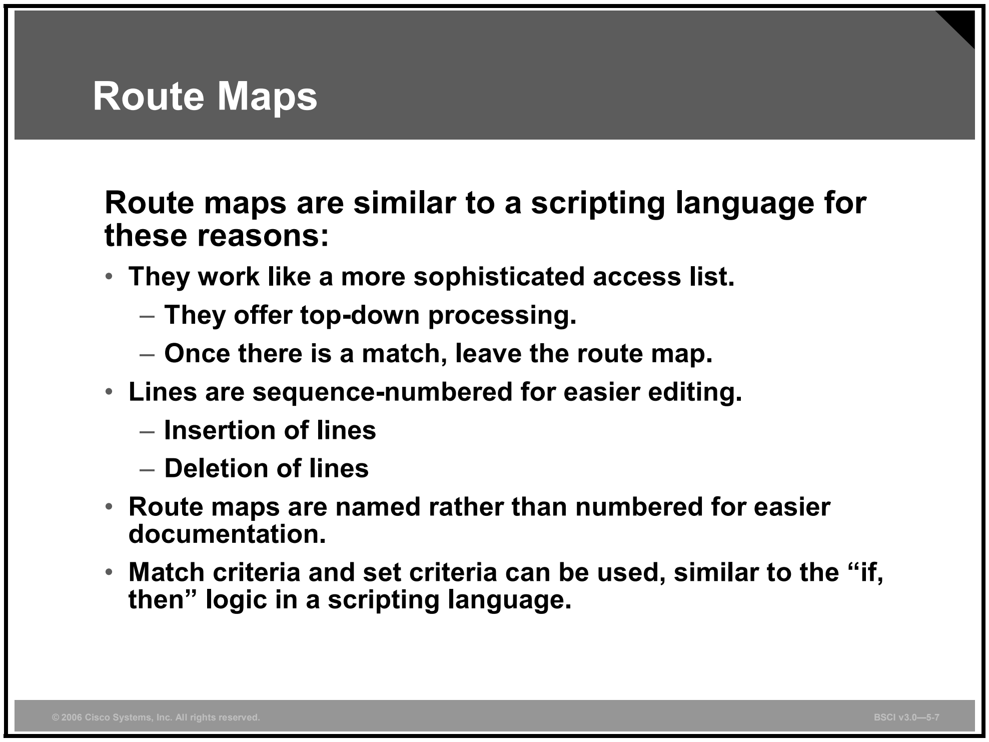
As shown in the figure, two-way redistribution is completed between RIP and OSPF. Networks 10.1.0.0 to 10.3.0.0 redistribute from RIP into OSPF. Route feedback could occur if another redistribution point is configured (router D) and OSPF then redistributes those networks back into RIP. ACL 2 allows the original RIP routes and denies all others. The distribute list configured under OSPF refers to this ACL.

The result is that networks 10.8.0.0 to 10.11.0.0, originated by OSPF, cannot be redistributed back into OSPF from RIP. Redistribution into RIP from OSPF is filtered with ACL 3. Router D will have a similar configuration to router B.

A distribute list hides network information, which could be considered a drawback in some circumstances. In a network with redundant paths, the goal of using a distribute list may be to prevent routing loops. The distribute list permits routing updates that enable only the desired paths to be advertised. Therefore, other routers in the network do not know about other ways to reach the filtered networks.

Defining Route Maps

Route maps are powerful and flexible configuration tools. This topic describes the functionality of route maps.



Route Maps

Route maps are similar to a scripting language for these reasons:

- **They work like a more sophisticated access list.**
  - **They offer top-down processing.**
  - **Once there is a match, leave the route map.**
- **Lines are sequence-numbered for easier editing.**
  - **Insertion of lines**
  - **Deletion of lines**
- **Route maps are named rather than numbered for easier documentation.**
- **Match criteria and set criteria can be used, similar to the "if, then" logic in a scripting language.**

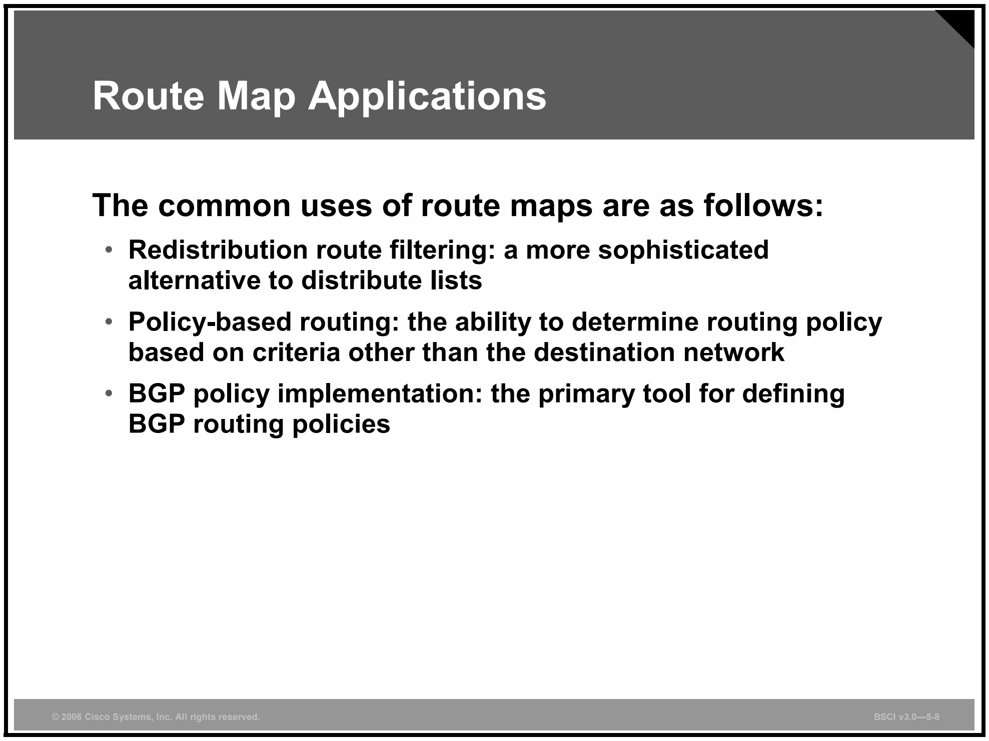
© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-7

Route maps are complex ACLs that allow conditions to be tested against a packet or route using the **match** commands. If the conditions match, then actions can be taken to modify attributes of the packet or route. These actions are specified by the **set** commands.

A collection of route map statements that have the same route map name is considered one route map. Within a route map, each route map statement is numbered and can be edited individually.

The statements in a route map are analogous to the lines of an ACL. Specifying the match conditions in a route map is similar to specifying the source and destination addresses and masks in an ACL.

One major difference between route maps and ACLs is that route maps can use the **set** commands to modify the packet or route.



Route Map Applications

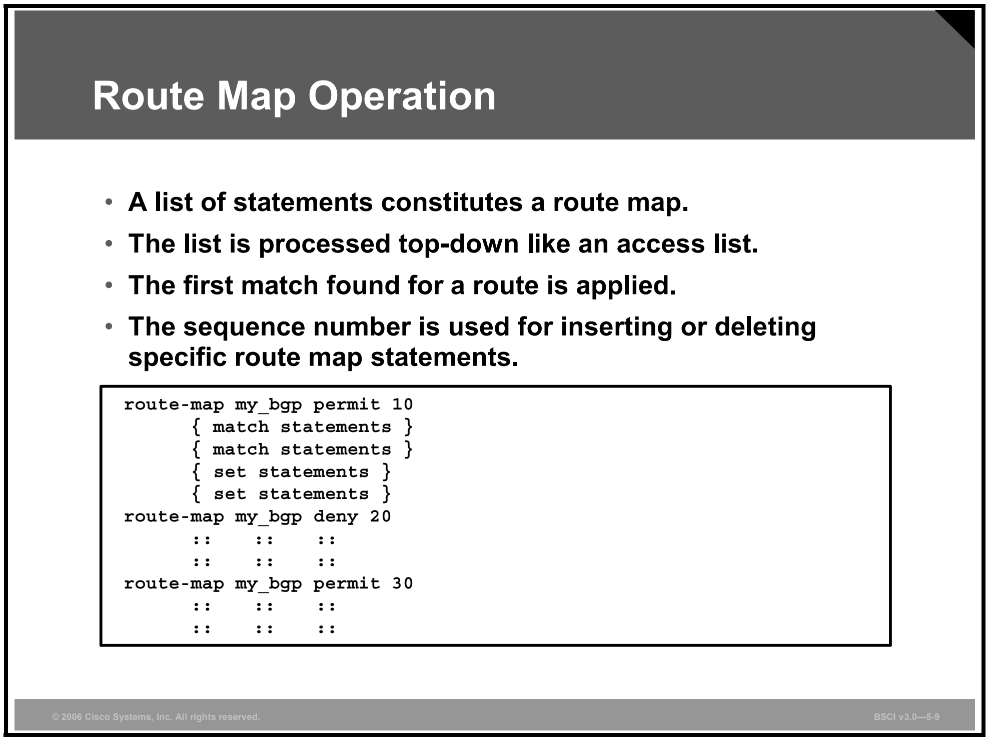
The common uses of route maps are as follows:

- **Redistribution route filtering: a more sophisticated alternative to distribute lists**
- **Policy-based routing: the ability to determine routing policy based on criteria other than the destination network**
- **BGP policy implementation: the primary tool for defining BGP routing policies**

BSCI v3.0—5-8

Network administrators use the route map tool for a variety of purposes. Several of the more common applications for route maps are as follows:

- **Route filtering during redistribution:** Redistribution nearly always requires some amount of route filtering. Although distribute lists can be used for this purpose, route maps offer an added benefit of manipulating routing metrics through the use of the **set** commands.
- **Policy-based routing (PBR):** Route maps can be used to match source and destination addresses, protocol types, and end-user applications. When a match occurs, a **set** command describes the interface or next-hop address to which the packet should be sent. PBR allows the operator to define routing policy other than basic destination-based routing using the routing table.
- **BGP:** Route maps are the primary tools for implementing Border Gateway Protocol (BGP) policy. Network administrators assign route maps to specific BGP sessions (neighbors) to control which routes are allowed to flow into and out of the BGP process. In addition to filtering, route maps provide sophisticated manipulation of BGP path attributes.



Route Map Operation

- A list of statements constitutes a route map.
- The list is processed top-down like an access list.
- The first match found for a route is applied.
- The sequence number is used for inserting or deleting specific route map statements.

```
route-map my_bgp permit 10
      { match statements }
      { match statements }
      { set statements }
      { set statements }
route-map my_bgp deny 20
      ::    ::    ::
      ::    ::    ::
route-map my_bgp permit 30
      ::    ::    ::
      ::    ::    ::
```

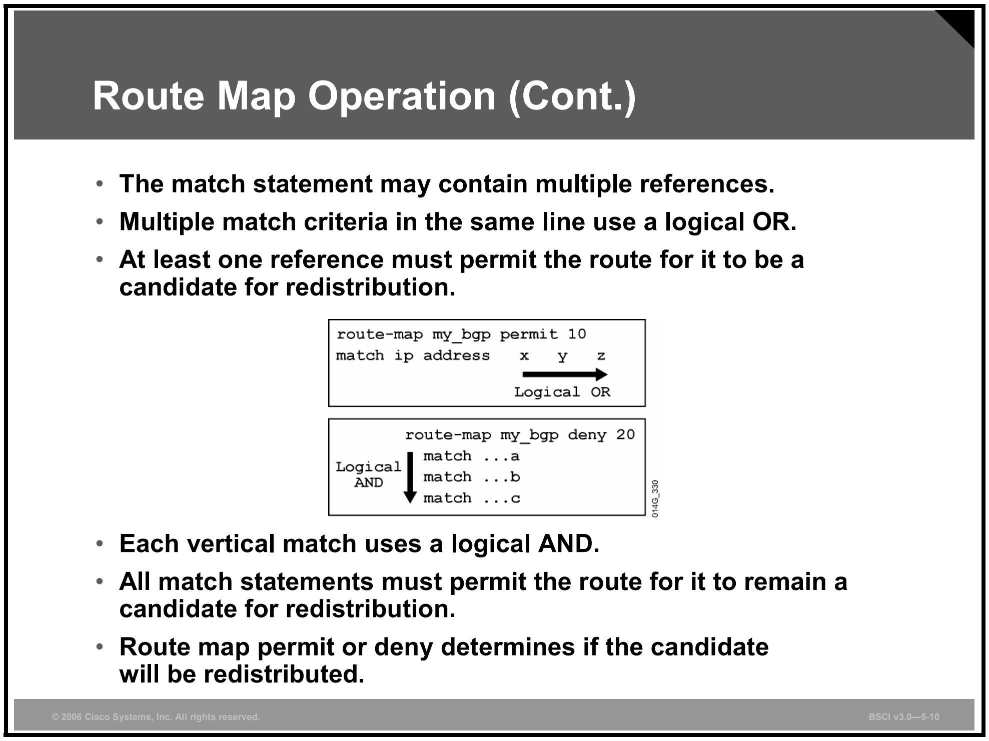
Route maps operate in a manner similar to ACLs. When determining which routes will be redistributed from one protocol to the next, the router checks each route against the route map, beginning with the top line.

Each line is sequence-numbered, both for top-down processing purposes and for editing purposes. Lines can be added or removed from a route map as changes are required.

Each line has a permit or deny statement. If a route is matched in the matching statements and the line statement is “permit,” then the router sets the metrics or other defined conditions and permits the redistribution of that route. The route map stops processing at the first match.

If the packet is matched and the route map line is “deny,” then the router stops at the matched line in the map and does not redistribute that route. Routes are filtered by this method.

Routes are checked from line to line looking for a match. If there is no match and the bottom of the route map is reached, then the router denies the route from being redistributed. There is always an implicit deny at the end of a route map.



Route Map Operation (Cont.)

- The match statement may contain multiple references.
- Multiple match criteria in the same line use a logical OR.
- At least one reference must permit the route for it to be a candidate for redistribution.

```
route-map my_bgp permit 10
match ip address   x   y   z
                  ———————————▶
                   Logical OR
```

```
         route-map my_bgp deny 20
        ┃ match ...a
Logical ┃ match ...b
  AND   ▼ match ...c
```

- Each vertical match uses a logical AND.
- All match statements must permit the route for it to remain a candidate for redistribution.
- Route map permit or deny determines if the candidate will be redistributed.

Matching statements in a route map can be complex. Multiple match criteria in the same line are processed with OR logic. Separate match criteria can also be applied vertically under a route map line. In this case, each match uses AND logic.

A route map may consist of multiple route map statements. The statements are processed top down, like an ACL. The first match found for a route is applied. The sequence number is used for inserting or deleting specific route map statements in a specific place in the route map.

The **match** route map configuration commands define the conditions to be checked. The **set** route map configuration commands define the actions that you should follow if there is a match.

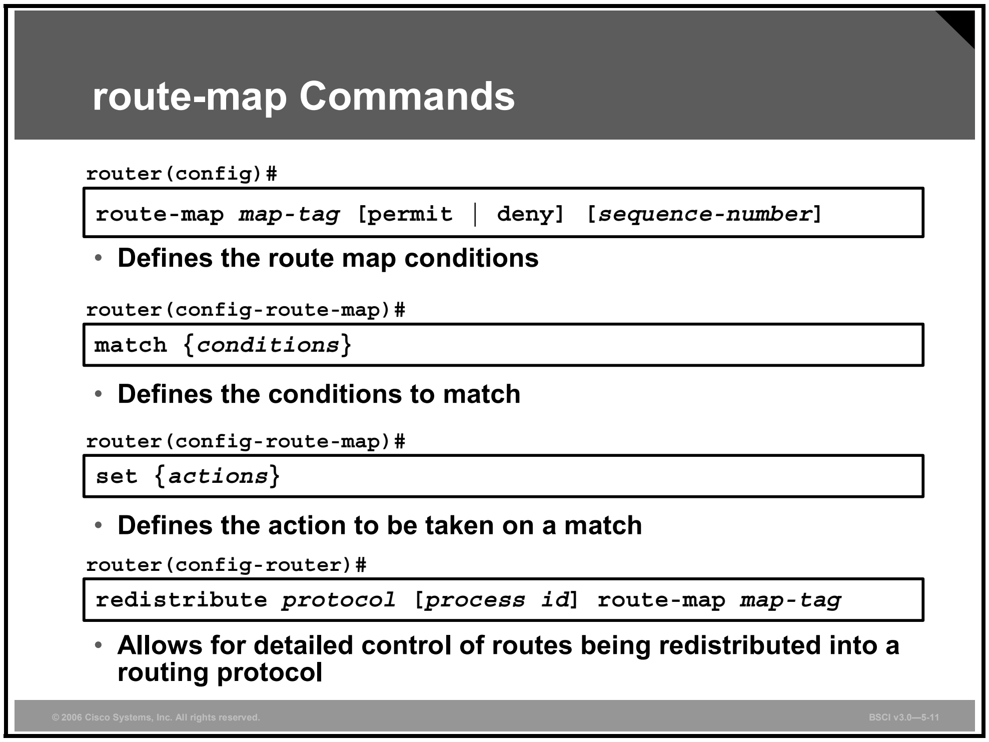
The single-match statement may contain multiple conditions. At least one condition in the match statement must be true to consider the statement a match (logical OR). A route map statement may contain multiple-match statements. All match statements in the route map statement must be true to consider the route map statement a match (logical AND).

The sequence number specifies the order in which conditions are checked. For example, if there are two statements in a route map named MYMAP, one with sequence 10 and the other with sequence 20, sequence 10 is checked first. If the match conditions in sequence 10 are not met, then sequence 20 is checked.

Like an ACL, there is an implicit deny any at the end of a route map. The consequences of this deny depend on how the route map is used.

Using route-map Commands

This topic describes how to use the **route-map** command to define the conditions for route filtering and redistribution.



route-map Commands

router(config)#

```
route-map map-tag [permit | deny] [sequence-number]
```

- **Defines the route map conditions**

router(config-route-map)#

```
match {conditions}
```

- **Defines the conditions to match**

router(config-route-map)#

```
set {actions}
```

- **Defines the action to be taken on a match**

router(config-router)#

```
redistribute protocol [process id] route-map map-tag
```

- **Allows for detailed control of routes being redistributed into a routing protocol**

BSCI v3.0—5-11

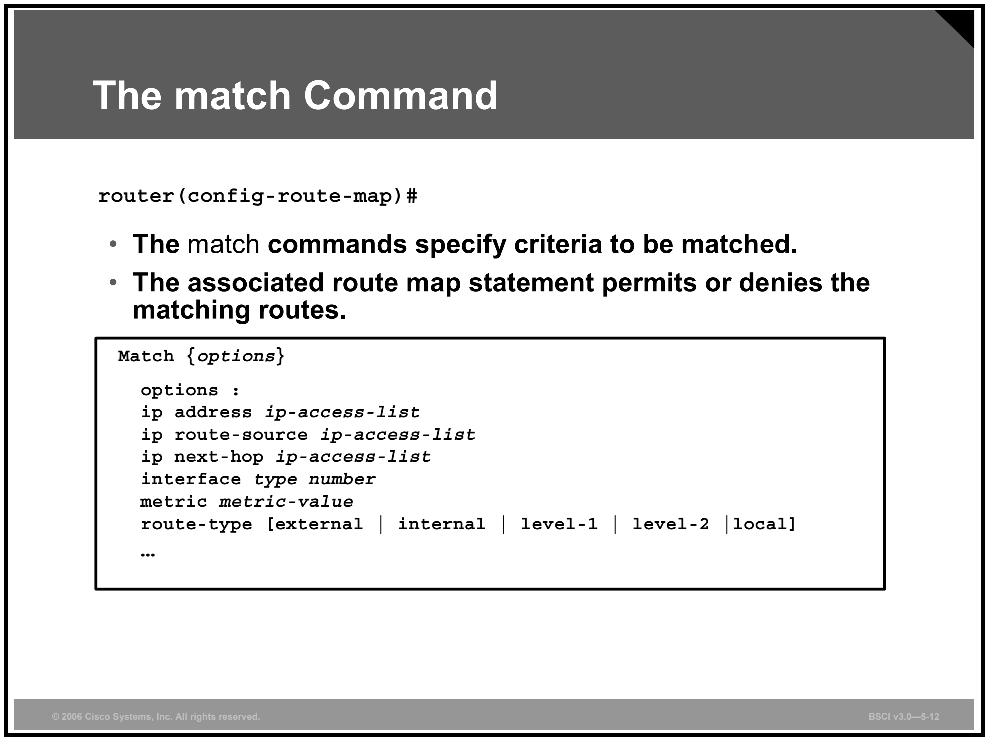
The **route-map** command is used to define the conditions for route filtering and redistribution.

The route-map Command Parameters

The table describes the parameters of the **route-map** command.

Parameter	Description
map-tag	Specifies the name of the route map
permit | deny	Specifies the action to be taken if the route map match conditions are met ■ permit = permit the matched route to be redistributed ■ deny = deny the matched route from being redistributed
sequence-number	Specifies the sequence number that indicates the position that a new route map statement will have in the list of route map statements already configured with the same route map name

When used for redistribution filtering, a route map is applied to the route redistribution process by adding the **route-map** command and *map-tag* to the end of the **redistribute** *protocol* command.



The match Command

```
router(config-route-map)#
```

- **The** match **commands specify criteria to be matched.**
- **The associated route map statement permits or denies the matching routes.**

```
Match {options}

  options :
  ip address ip-access-list
  ip route-source ip-access-list
  ip next-hop ip-access-list
  interface type number
  metric metric-value
  route-type [external | internal | level-1 | level-2 |local]
  …
```

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-12

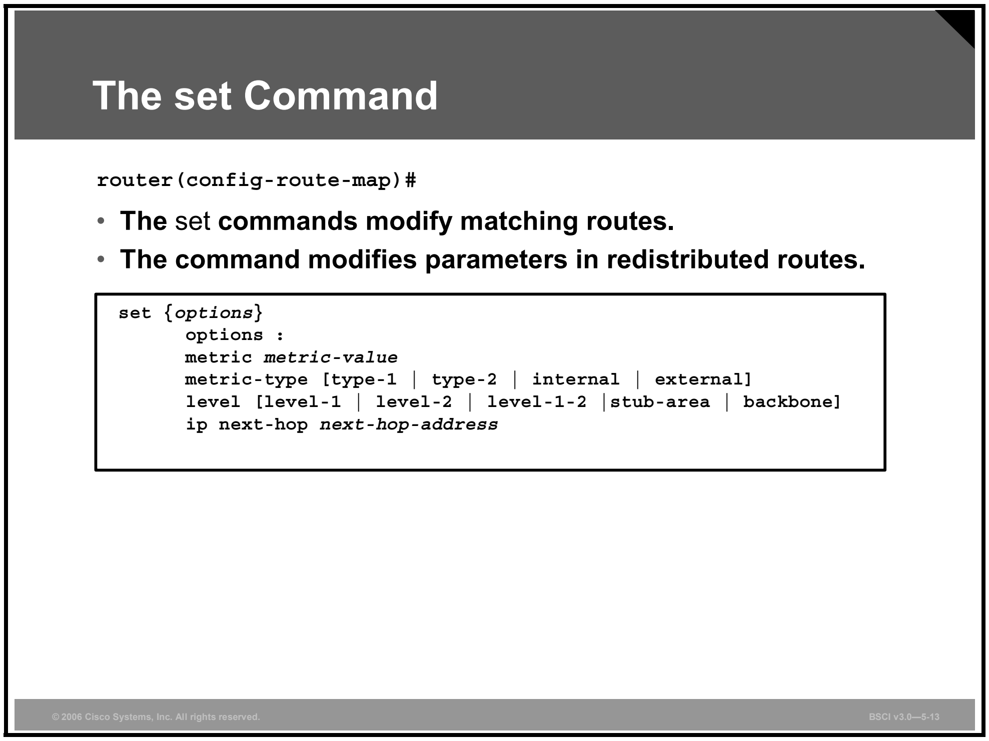
The **match** command is applied within a route map.

The match Commands

The table lists some of the variety of match criteria that can be defined.

Command	Description
match community	Matches a BGP community
match interface	Matches any routes that have the next hop out of one of the interfaces specified
match ip address	Matches any routes that have a destination network number address that is permitted by a standard or extended ACL
match ip next-hop	Matches any routes that have a next-hop router address that is passed by one of the ACLs specified
match ip route-source	Matches routes that have been advertised by routers and access servers at the address that is specified by the ACLs
match length	Matches based on the Layer 3 length of a packet
match metric	Matches routes with the metric specified
match route-type	Matches routes of the specified type
match tag	Matches tag of a route

The table presents a general list of match criteria. Some criteria are used for BGP policy, some criteria are used for PBR, and some criteria are used for redistribution filtering.



The set Command

```
router(config-route-map)#
```

- **The** set **commands modify matching routes.**
- **The command modifies parameters in redistributed routes.**

```
set {options}
      options :
      metric metric-value
      metric-type [type-1 | type-2 | internal | external]
      level [level-1 | level-2 | level-1-2 |stub-area | backbone]
      ip next-hop next-hop-address
```

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—5-13

The **set** commands are used within a route map to change or add characteristics, such as metrics, to any routes that have met a match criterion.

The set Commands

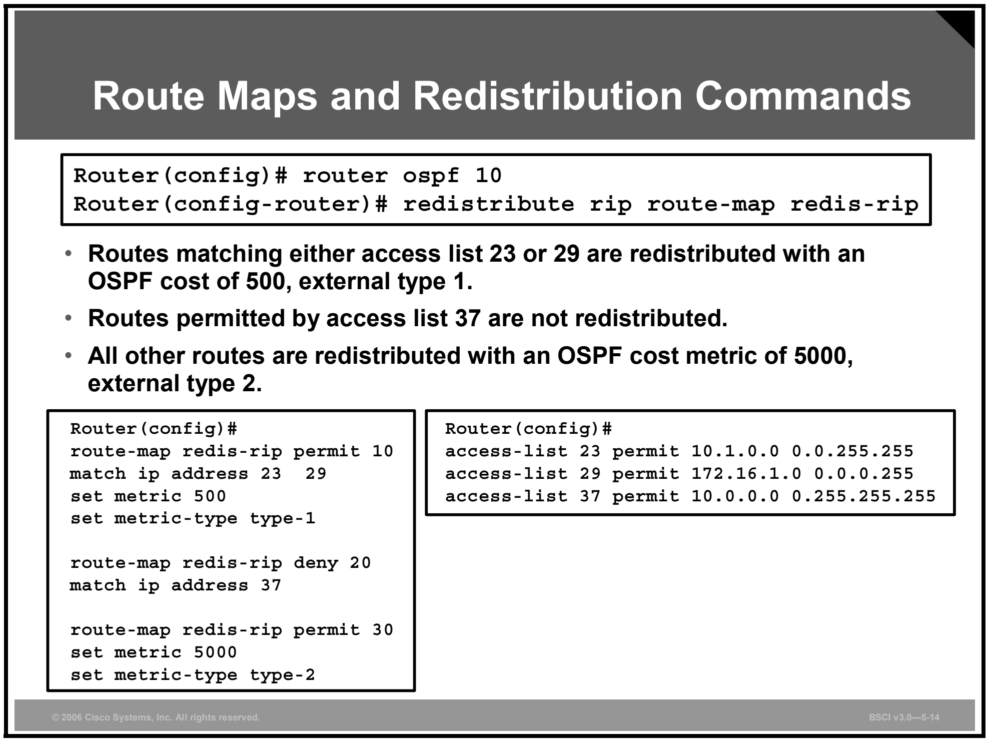
The table lists some of the **set** options available.

Command	Description
`set as-path`	Modifies an AS path for BGP routes
`set automatic-tag`	Computes automatically the tag value
`set community`	Sets the BGP communities attribute
`set default interface`	Indicates where to output packets that pass a match clause of a route map for policy routing and have no explicit route to the destination
`set interface`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set ip default next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing and for which Cisco IOS software has no explicit route to a destination
`set ip next-hop`	Indicates where to output packets that pass a match clause of a route map for policy routing
`set level`	Indicates where to import routes for IS-IS and OSPF
`set local-preference`	Specifies a BGP local preference value
`set metric`	Sets the metric value for a routing protocol
`set metric-type`	Sets the metric type for the destination routing protocol
`set tag`	Sets tag value for destination routing protocol
`set weight`	Specifies the BGP weight value

Not all the **set** options that are listed here are used for redistribution purposes. The table includes options for BGP and PBR.

Implementing Route Maps with Redistribution

This topic describes how to implement route maps with route redistribution.



Route Maps and Redistribution Commands

```
Router(config)# router ospf 10
Router(config-router)# redistribute rip route-map redis-rip
```

- **Routes matching either access list 23 or 29 are redistributed with an OSPF cost of 500, external type 1.**
- **Routes permitted by access list 37 are not redistributed.**
- **All other routes are redistributed with an OSPF cost metric of 5000, external type 2.**

```
Router(config)#
route-map redis-rip permit 10
match ip address 23  29
set metric 500
set metric-type type-1

route-map redis-rip deny 20
match ip address 37

route-map redis-rip permit 30
set metric 5000
set metric-type type-2
```

```
Router(config)#
access-list 23 permit 10.1.0.0 0.0.255.255
access-list 29 permit 172.16.1.0 0.0.0.255
access-list 37 permit 10.0.0.0 0.255.255.255
```

Example: Route Maps and Redistribution Commands

In this example, RIPv1 is being redistributed into OSPF 10. A route map called “redis-rip” has been attached to the **redistribute rip** command.

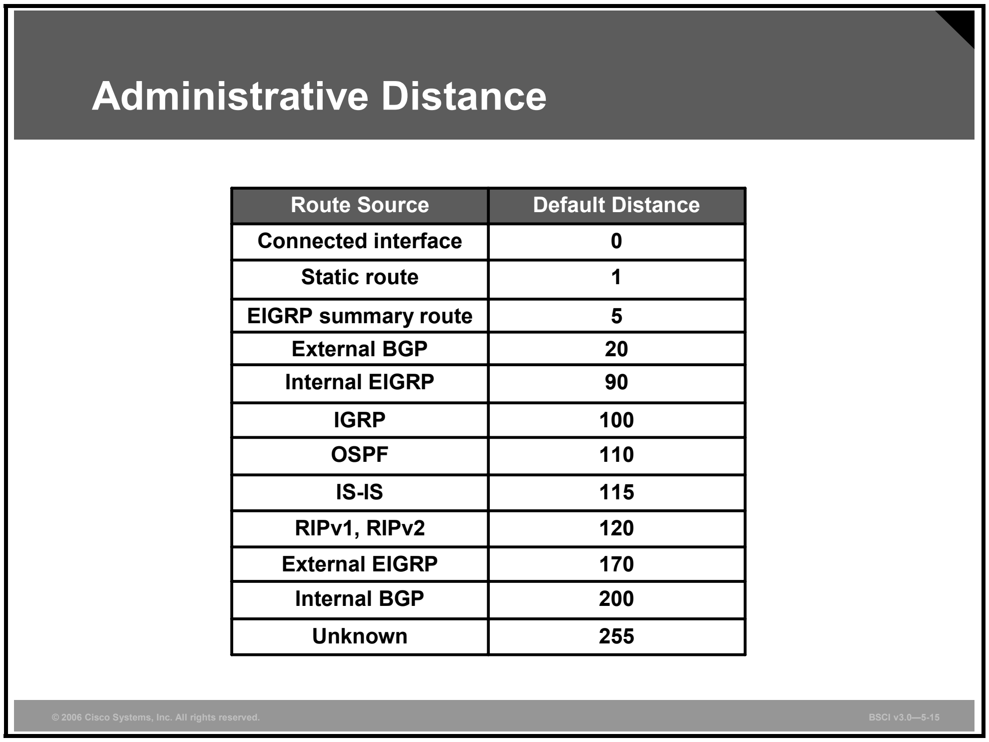
Sequence number 10 of the route map is looking for an IP address match in ACL 23 or ACL 29. If a match is found, then the router redistributes the route into OSPF with a cost metric of 500 and sets the new OSPF route to external type 1.

If there is no match to line 10, move to line 20. If there is a match in ACL 37, then do not let that route redistribute into OSPF because sequence number 20 is a deny.

If there is no match to sequence number 20, move to 30. Because 30 is a permit and there is no match criterion, all remaining routes are redistributed into OSPF with a cost metric of 5000 and an external metric of type 2.

Defining Administrative Distance

Cisco Systems routers use administrative distance when using more than one routing protocol. This topic describes the features of administrative distance in terms of routing protocols.



Administrative Distance

Route Source	Default Distance
Connected interface	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIPv1, RIPv2	120
External EIGRP	170
Internal BGP	200
Unknown	255

BSCI v3.0—5-15

Most routing protocols have metric structures and algorithms that are not compatible with other protocols. It is critical for a network using multiple routing protocols to have seamless exchange of route information and the ability to select the best path across multiple protocols.

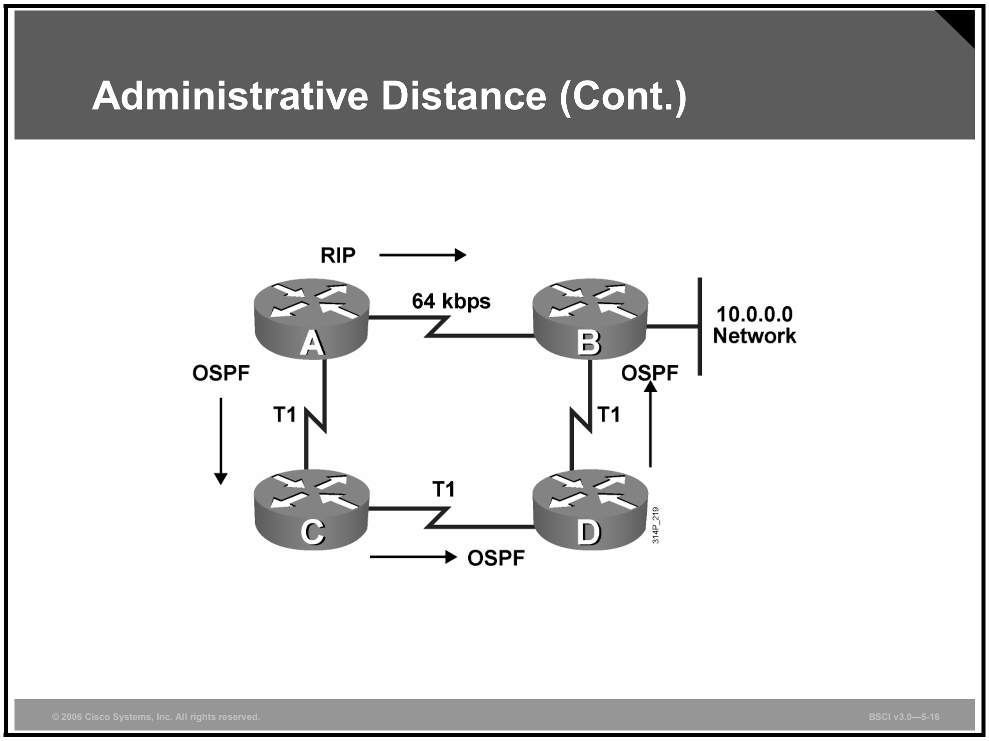
Cisco routers use a value called administrative distance to select the best path when they learn two or more routes to the same destination from different routing protocols. Administrative distance rates the *believability* of a routing protocol. Cisco has assigned a default administrative distance value to each routing protocol supported on its routers.

Each routing protocol is prioritized in the order of most believable to least believable. Some examples of prioritization are as follows:

- Prefer manually configured routes (static routes) to dynamically learned routes
- Prefer protocols with sophisticated metrics to protocols with more deterministic metrics
- Prefer External Border Gateway Protocol (EBGP) to most other dynamic protocols

In the figure, the table lists the default administrative distance of the protocols supported by Cisco routers. The administrative distance is a value between 0 and 255. The lower the administrative distance value, the higher the reliability of the protocol.

Note	IGRP is no longer supported, as of Cisco IOS Software Release 12.3.



Administrative Distance (Cont.)

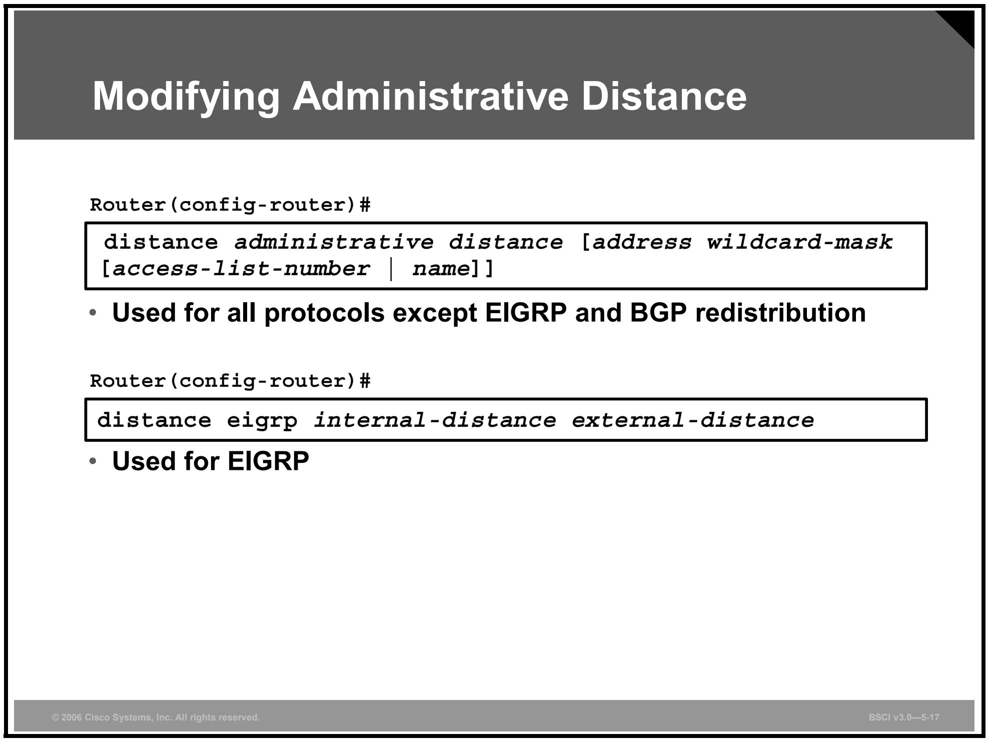
Example: Administrative Distance

For example, if router A receives a route to network 10.0.0.0 from RIP and receives a route to the same network from OSPF, the router compares the administrative distance of RIP, 120, with the administrative distance of OSPF, 110.

The router uses the administrative distance value to determine that OSPF is more reliable and adds the OSPF version of the route to the routing table.

Modifying Administrative Distance

This topic describes how to globally modify administrative distance on the router for a particular routing protocol or specifically for certain routes.



Modifying Administrative Distance

```
Router(config-router)#
```

```
distance administrative distance [address wildcard-mask
[access-list-number | name]]
```

- **Used for all protocols except EIGRP and BGP redistribution**

```
Router(config-router)#
```

```
distance eigrp internal-distance external-distance
```

- **Used for EIGRP**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-17

In some cases, a router selects a suboptimal path if it believes a routing protocol with a better administrative distance, even though it is actually a routing protocol with a worse route.

Assigning an undesired routing protocol a larger administrative distance ensures that routers select routes from the desired routing protocol. The figure illustrates the commands for changing the default administrative distance.

The **distance** command can be used to change the default administrative distance for all protocols except EIGRP and BGP.

The distance Command Parameters

The table explains each field in the **distance** command

Parameter	Description
administrative distance	Sets the administrative distance. An integer from 1 to 255. Routes with a distance of 255 are not installed in the routing table. A value of 0 is reserved for directly connected networks.
address	(Optional) Specifies the IP address and filters networks according to the IP address of the router supplying the routing information.
wildcard-mask	(Optional) Specifies the wildcard mask for an IP address. A bit set to 1 in the mask argument instructs the software to ignore the corresponding bit in the address value. Use an address/mask of 0.0.0.0 255.255.255.255 to match any IP address (any source router supplying the routing information).
access-list-number | name	(Optional) Specifies the number or name of the standard ACL to apply to incoming routing updates. It allows filtering of the advertised networks.

For EIGRP, use the **distance eigrp** command. EIGRP assigns different administrative distance values to routes learned natively through EIGRP and to routes redistributed in from other sources.

By default, natively learned routes have an administrative distance of 90, but external routes have an administrative distance of 170.

The distance eigrp Command Parameters

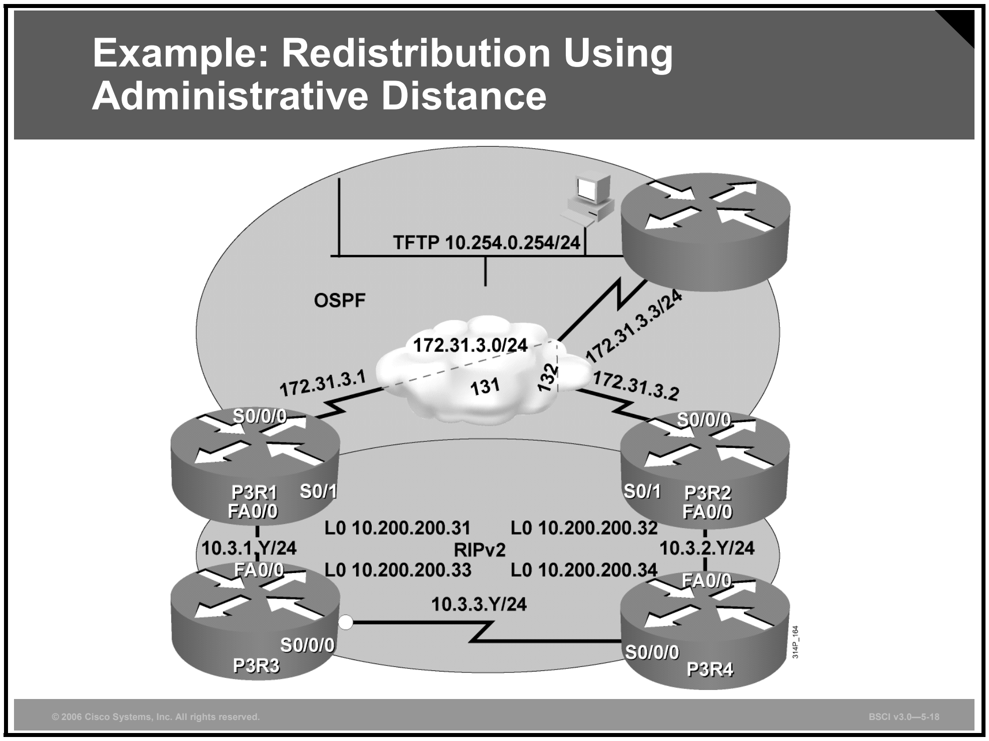
The table explains each field in the **distance eigrp** command.

Parameter	Description
external-distance	Sets the administrative distance for EIGRP external routes. External routes are routes for which the best path is learned from a neighbor external to the AS.
internal-distance	Specifies the administrative distance for EIGRP internal routes. Internal routes are routes that are learned from another entity within the AS.

For BGP, use the **distance bgp** command. BGP assigns different administrative distance values to routes learned through Internal Border Gateway Protocol (IBGP) and routes learned through External Border Gateway Protocol (EBGP).

Defining the Impact of Administrative Distance Changes

This topic describes the impact of administrative distance changes on routing tables.



Example: Redistribution Using Administrative Distance

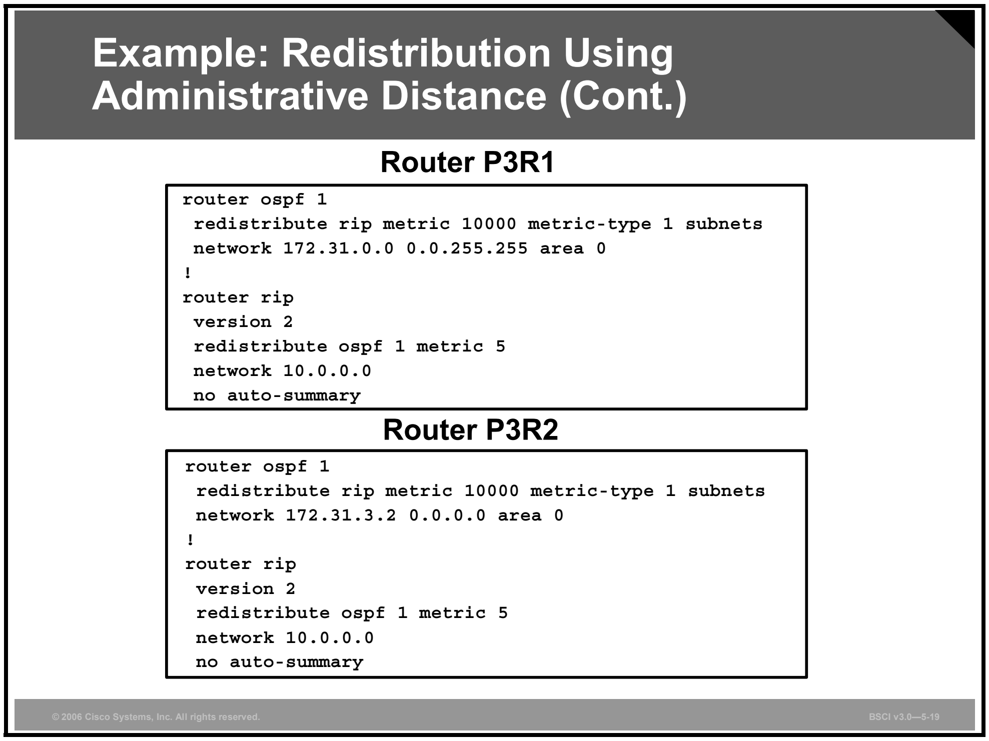
BSCI v3.0—5-18

The following examples describe a network using multiple routing protocols. There are a number of ways to correct path selection problems in a redistribution environment. These examples show how a problem can occur, where it appears, and one possible way to resolve it.

Example: Redistribution Using Administrative Distance

The figure illustrates a network with RIP and OSPF routing domains. Recall that OSPF is more believable than RIP because it has an administrative distance of 110, and RIP has an administrative distance of 120.

If, for example, the boundary router (P3R1 or P3R2) learns about network 10.3.3.0 via RIPv2 and also via OSPF, the OSPF route is used and inserted into the routing table because OSPF has a lower administrative distance than RIPv2, even though the path via OSPF might be the longer (worse) path.



Example: Redistribution Using Administrative Distance (Cont.)

Router P3R1

```
router ospf 1
 redistribute rip metric 10000 metric-type 1 subnets
 network 172.31.0.0 0.0.255.255 area 0
!
router rip
 version 2
 redistribute ospf 1 metric 5
 network 10.0.0.0
 no auto-summary
```

Router P3R2

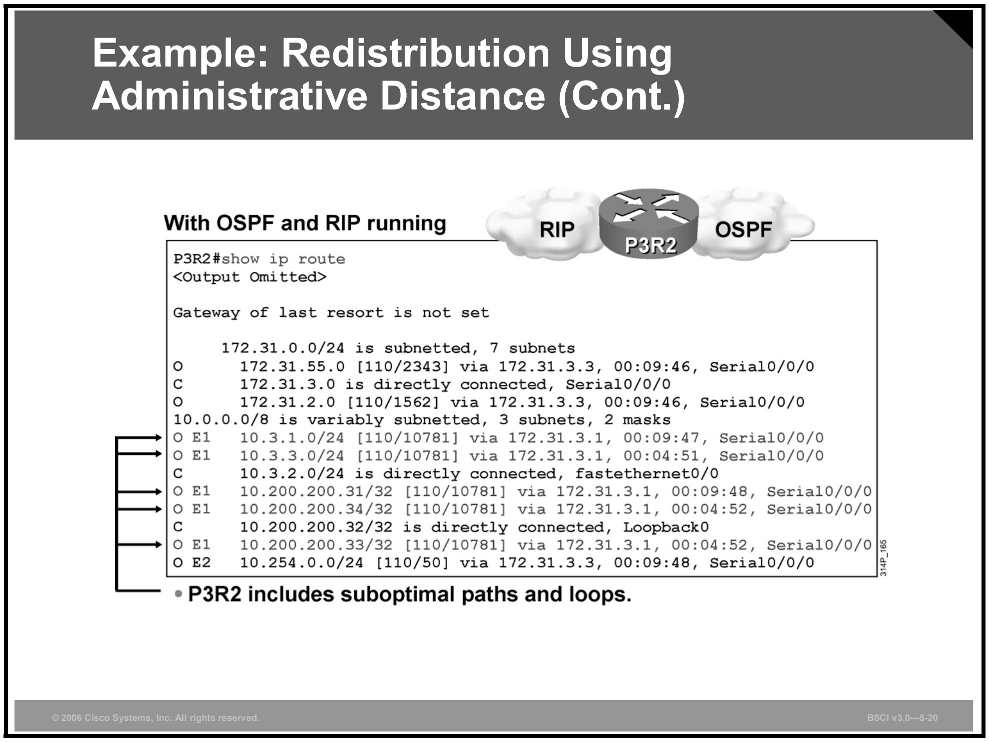
```
router ospf 1
 redistribute rip metric 10000 metric-type 1 subnets
 network 172.31.3.2 0.0.0.0 area 0
!
router rip
 version 2
 redistribute ospf 1 metric 5
 network 10.0.0.0
 no auto-summary
```

Example: Configurations for the P3R1 and P3R2 Routers

The figure illustrates the configurations for the P3R1 and P3R2 routers. These configurations redistribute RIP into OSPF and OSPF into RIP on both routers.

The redistribution into OSPF sets a default OSPF metric of 10000 to make these routes less preferred than native OSPF routes and protect against route feedback. The redistribute statement also sets the metric type to E1 so that the route metrics continue to accrue, and the router redistributes subnet information.

The redistribution into RIP sets a default RIP metric of 5 to also protect against route feedback.



Example: Redistribution Using Administrative Distance (Cont.)

With OSPF and RIP running

```
P3R2#show ip route
<Output Omitted>

Gateway of last resort is not set

     172.31.0.0/24 is subnetted, 7 subnets
O       172.31.55.0 [110/2343] via 172.31.3.3, 00:09:46, Serial0/0/0
C       172.31.3.0 is directly connected, Serial0/0/0
O       172.31.2.0 [110/1562] via 172.31.3.3, 00:09:46, Serial0/0/0
10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
O E1    10.3.1.0/24 [110/10781] via 172.31.3.1, 00:09:47, Serial0/0/0
O E1    10.3.3.0/24 [110/10781] via 172.31.3.1, 00:04:51, Serial0/0/0
C       10.3.2.0/24 is directly connected, fastethernet0/0
O E1    10.200.200.31/32 [110/10781] via 172.31.3.1, 00:09:48, Serial0/0/0
O E1    10.200.200.34/32 [110/10781] via 172.31.3.1, 00:04:52, Serial0/0/0
C       10.200.200.32/32 is directly connected, Loopback0
O E1    10.200.200.33/32 [110/10781] via 172.31.3.1, 00:04:52, Serial0/0/0
O E2    10.254.0.0/24 [110/50] via 172.31.3.3, 00:09:48, Serial0/0/0
```

- P3R2 includes suboptimal paths and loops.

BSCI v3.0—5-20

Example: Routing Table After Redistribution

This figure displays the routing table on the P3R2 router after redistribution has occurred. The P3R2 router learned RIP and OSPF routes but lists only OSPF routes in the routing table.

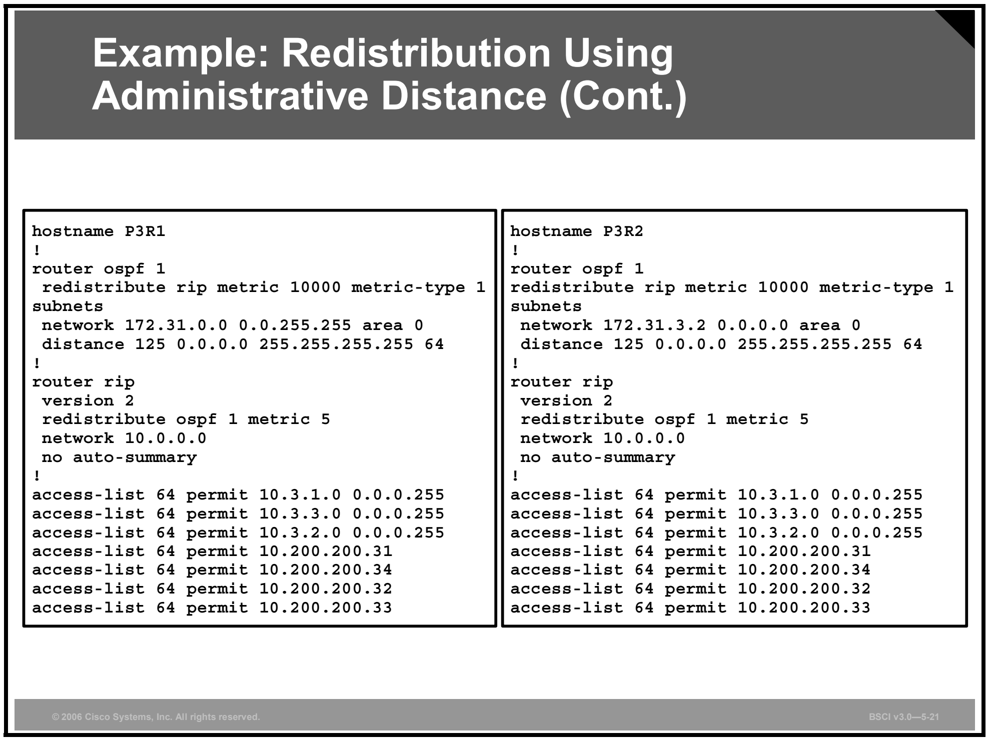
The first edge router to set up redistribution has a normal routing table and retains the RIP routes. The second edge router chooses the OSPF routes over its RIP routes. The paths to the internal RIP routes are shown as going through the core because of the dual mutual redistribution points.

OSPF is informed about the RIP routes via redistribution. OSPF then advertises the RIP routes via OSPF routes to its neighboring router. The neighbor router is also informed about the same routes via RIP. However, OSPF has a better administrative distance than RIP, so the RIP routes are not put into the routing table.

OSPF was configured on the P3R1 router first, and P3R2 then received information about the internal (native RIP) routes from both OSPF and RIP. It prefers the OSPF routes because OSPF has a lower administrative distance. Therefore, none of the RIP routes appear in the table.

Refer back to the topology diagram to trace some of the routes. The redistribution has resulted in suboptimal paths to many of the networks.

For instance, 10.200.200.34 is a loopback interface on router P3R4. P3R4 is directly attached to P3R2. However, the OSPF path to that loopback interface goes through P3R1, then P3R3, then P3R4 before it reaches its destination. The OSPF path taken is actually a longer (worse) path than the more direct RIP path.



Example: Redistribution Using Administrative Distance (Cont.)

```
hostname P3R1
!
router ospf 1
 redistribute rip metric 10000 metric-type 1
subnets
 network 172.31.0.0 0.0.255.255 area 0
 distance 125 0.0.0.0 255.255.255.255 64
!
router rip
 version 2
 redistribute ospf 1 metric 5
 network 10.0.0.0
 no auto-summary
!
access-list 64 permit 10.3.1.0 0.0.0.255
access-list 64 permit 10.3.3.0 0.0.0.255
access-list 64 permit 10.3.2.0 0.0.0.255
access-list 64 permit 10.200.200.31
access-list 64 permit 10.200.200.34
access-list 64 permit 10.200.200.32
access-list 64 permit 10.200.200.33
```

```
hostname P3R2
!
router ospf 1
redistribute rip metric 10000 metric-type 1
subnets
 network 172.31.3.2 0.0.0.0 area 0
 distance 125 0.0.0.0 255.255.255.255 64
!
router rip
 version 2
 redistribute ospf 1 metric 5
 network 10.0.0.0
 no auto-summary
!
access-list 64 permit 10.3.1.0 0.0.0.255
access-list 64 permit 10.3.3.0 0.0.0.255
access-list 64 permit 10.3.2.0 0.0.0.255
access-list 64 permit 10.200.200.31
access-list 64 permit 10.200.200.34
access-list 64 permit 10.200.200.32
access-list 64 permit 10.200.200.33
```

BSCI v3.0—5-21

One of the boundary routers (P3R2 in this example) selected the poor paths because OSPF has a better administrative distance than RIP. You can change the administrative distance of the redistributed RIP routes to ensure that the boundary routers select the native RIP routes, as illustrated in the figure.

The **distance** command modifies the administrative distance of the OSPF routes to the networks that match ACL 64.

The distance Command Parameters

The table describes some of the command parameters used in the example.

Parameter	Description
`125`	Defines the administrative distance that specified routes will be assigned

`0.0.0.0`
`255.255.255.255`
	Defines the source address of the router supplying the routing information—in this case, any router

`64`	Defines the ACL to be used to filter incoming routing updates to determine which will have their administrative distance changed

ACL 64 is used to match all the native RIP routes. The **access-list 64 permit 10.3.1.0** command configures a standard ACL to permit the 10.3.1.0 network. Other similar access-list statements permit the other internal native RIP networks.

The access-list Parameters

The table describes some of the command parameters used in the example

Parameter	Description
`64`	Displays the ACL number
`permit`	Allows all networks that match the address to be permitted, in this case to have their administrative distance changed
`10.3.1.0`	Displays a network to be permitted, in this case, to have its administrative distance changed

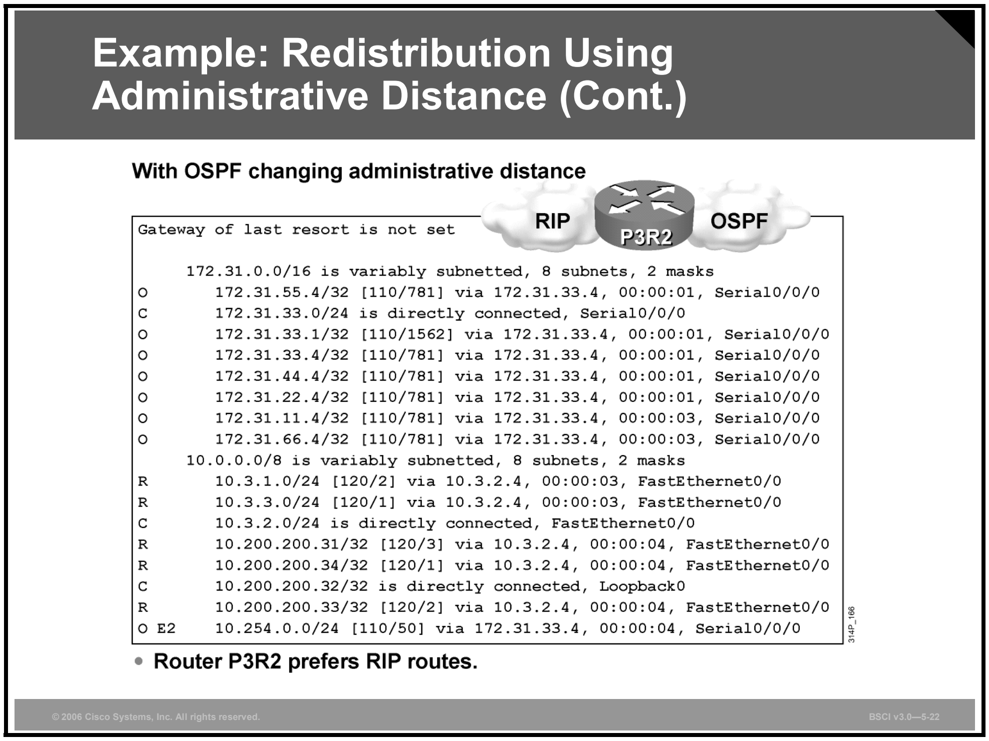
In the preceding figure, both of the redistributing routers are configured to assign an administrative distance of 125 to OSPF routes that are advertised for the networks that are listed in ACL 64.

ACL 64 has permit statements for the internal native RIP networks of 10.3.1.0, 10.3.2.0, and 10.3.3.0, as well as the loopback networks of 10.200.200.31, 10.200.200.32, 10.200.200.33, and 10.200.200.34.

When either one of the redistributing routers learns about these networks from RIP, it selects the routes learned from RIP (with a lower administrative distance of 120) over the same routes learned from OSPF (with an administrative distance of 125), and puts only the RIP routes in the routing table.

Note that the **distance** command is part of the OSPF routing process configuration because the administrative distance should be changed for these routes when they are advertised by OSPF, not by RIP.

You need to configure the **distance** command on both redistributing routers because either one can have suboptimal routes, depending on which redistributing router sends the OSPF updates about the RIP networks to the other redistributing router first.



Example: Redistribution Using Administrative Distance (Cont.)

With OSPF changing administrative distance

RIP
P3R2
OSPF

```
Gateway of last resort is not set

     172.31.0.0/16 is variably subnetted, 8 subnets, 2 masks
O       172.31.55.4/32 [110/781] via 172.31.33.4, 00:00:01, Serial0/0/0
C       172.31.33.0/24 is directly connected, Serial0/0/0
O       172.31.33.1/32 [110/1562] via 172.31.33.4, 00:00:01, Serial0/0/0
O       172.31.33.4/32 [110/781] via 172.31.33.4, 00:00:01, Serial0/0/0
O       172.31.44.4/32 [110/781] via 172.31.33.4, 00:00:01, Serial0/0/0
O       172.31.22.4/32 [110/781] via 172.31.33.4, 00:00:01, Serial0/0/0
O       172.31.11.4/32 [110/781] via 172.31.33.4, 00:00:03, Serial0/0/0
O       172.31.66.4/32 [110/781] via 172.31.33.4, 00:00:03, Serial0/0/0
     10.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
R       10.3.1.0/24 [120/2] via 10.3.2.4, 00:00:03, FastEthernet0/0
R       10.3.3.0/24 [120/1] via 10.3.2.4, 00:00:03, FastEthernet0/0
C       10.3.2.0/24 is directly connected, FastEthernet0/0
R       10.200.200.31/32 [120/3] via 10.3.2.4, 00:00:04, FastEthernet0/0
R       10.200.200.34/32 [120/1] via 10.3.2.4, 00:00:04, FastEthernet0/0
C       10.200.200.32/32 is directly connected, Loopback0
R       10.200.200.33/32 [120/2] via 10.3.2.4, 00:00:04, FastEthernet0/0
O E2    10.254.0.0/24 [110/50] via 172.31.33.4, 00:00:04, Serial0/0/0
```

- **Router P3R2 prefers RIP routes.**

The output shows that router P3R2 now retains the more direct paths to the internal networks by learning them from RIP. However, some routing information is lost with this configuration. For example, depending on the actual bandwidths, the OSPF path may have been better for the 10.3.1.0 network. It may have made sense not to include 10.3.1.0 in the ACL.

Example: Knowing Your Network

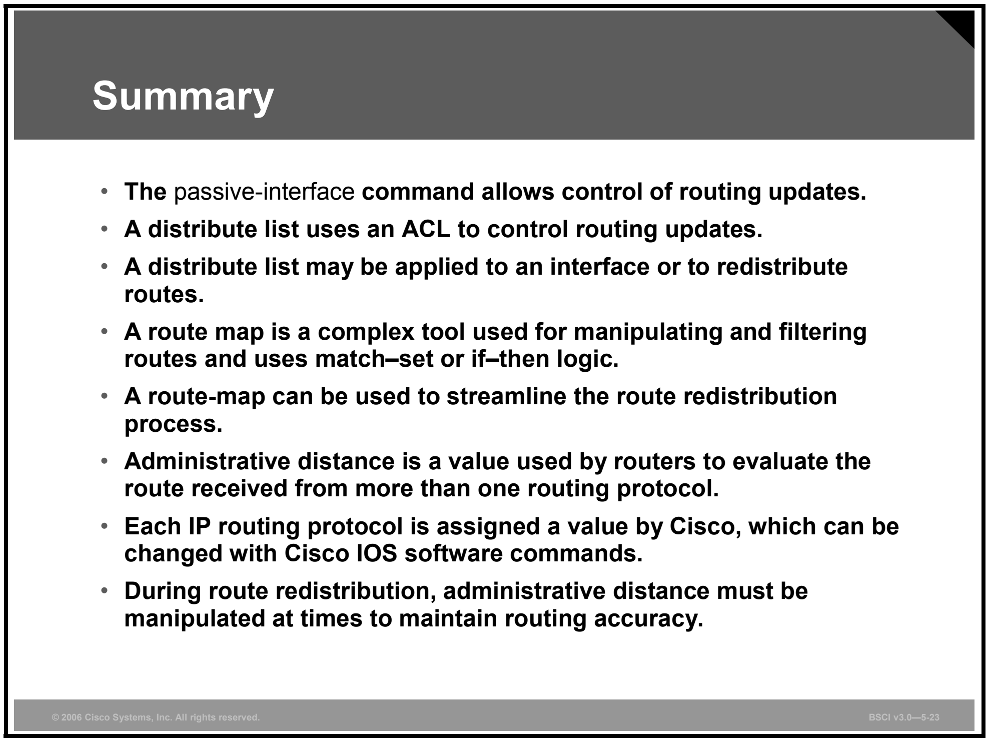
This example illustrates the importance of knowing your network before you implement redistribution and of closely examining which routes the routers are selecting after redistribution is enabled.

Pay particular attention to routers that can select from a number of possible redundant paths to a network, because they are more likely to select suboptimal paths.

The most important feature of using administrative distance to control route preference is that no path information is lost; the OSPF information is still in the OSPF database. If the primary path is lost, the OSPF path can reassert itself, and the router will maintain connectivity with those networks.

Summary

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



Summary

- **The** passive-interface **command allows control of routing updates.**
- **A distribute list uses an ACL to control routing updates.**
- **A distribute list may be applied to an interface or to redistribute routes.**
- **A route map is a complex tool used for manipulating and filtering routes and uses match–set or if–then logic.**
- **A route-map can be used to streamline the route redistribution process.**
- **Administrative distance is a value used by routers to evaluate the route received from more than one routing protocol.**
- **Each IP routing protocol is assigned a value by Cisco, which can be changed with Cisco IOS software commands.**
- **During route redistribution, administrative distance must be manipulated at times to maintain routing accuracy.**

Lesson 4

Implementing Advanced Cisco IOS Features: Configuring DHCP

Overview

DHCP is used to provide dynamic IP address allocation to TCP/IP hosts and Cisco Systems devices. It utilizes a client/server model, and the DHCP server can be a Windows server, a UNIX-based server, or a Cisco IOS device.

Configuring Cisco IOS devices as DHCP servers, DHCP relay agents, and DHCP clients allows a network administrator to implement more options for DHCP and also to implement levels of DHCP service for a more robust and efficient network solution.

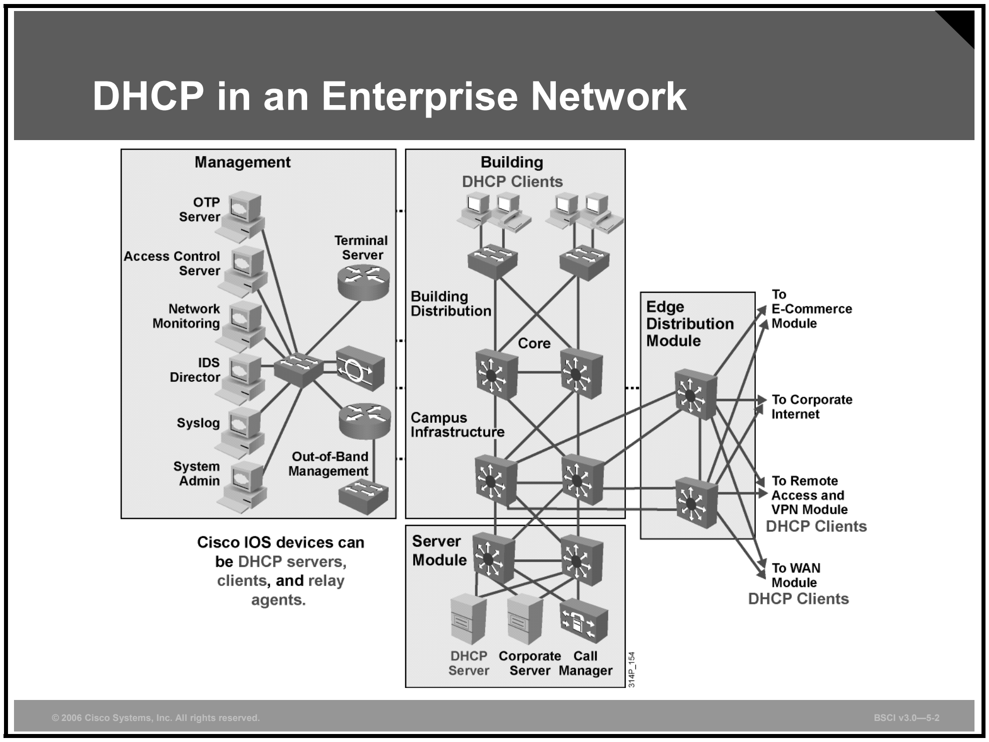
Objectives

Upon completing this lesson, you will be able to describe and configure DHCP services. This ability includes being able to meet these objectives:

- Describe the function and purpose of DHCP
- Explain how to enable the DHCP server on a Cisco IOS device
- Explain how to configure DHCP options
- Explain how to enable DHCP relay
- Explain how to configure a Cisco IOS Ethernet interface as a DHCP client
- Explain why and how to control broadcast packets and which protocols are forwarded with an IP helper address

Describing the Purpose of DHCP

This topic describes where DHCP functions occur in an enterprise network.



DHCP in an Enterprise Network

Cisco IOS devices can be DHCP servers, clients, and relay agents.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-2

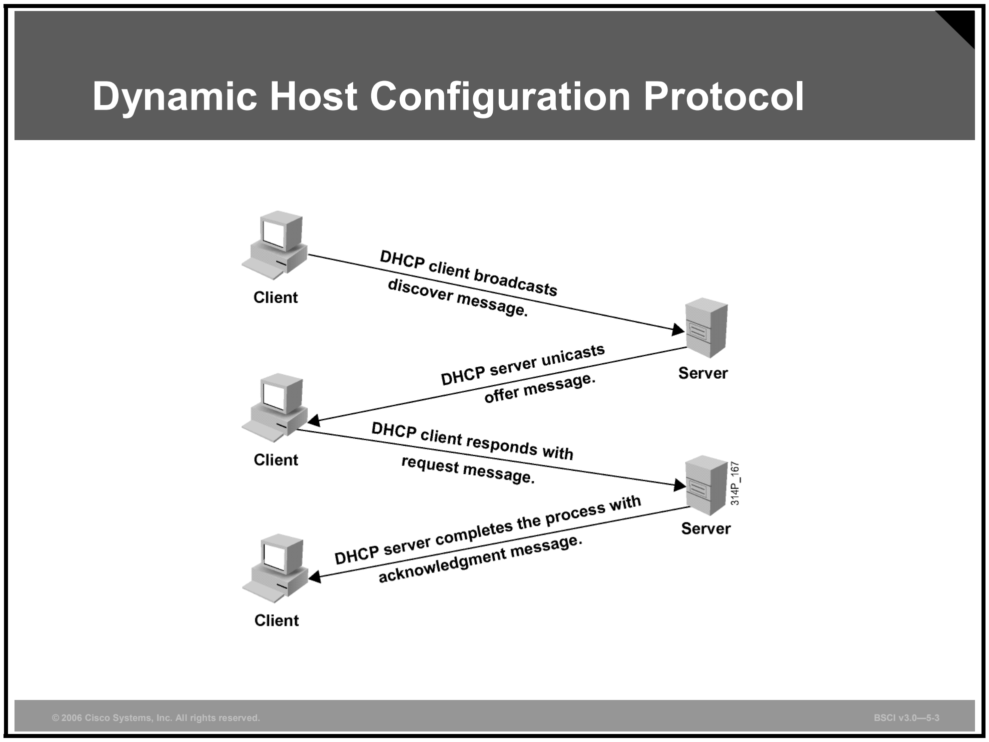
DHCP is structured on the Bootstrap Protocol (BOOTP) server and BOOTP well-known ports in User Datagram Protocol (UDP). Previous to DHCP, IP addresses were manually administered to IP hosts, which was a tedious, error-prone, and labor-intensive process.

DHCP allows IP addresses to be automatically assigned to DHCP clients. The DHCP service can be implemented with a server or with a Cisco IOS device.

The figure shows where DHCP would be implemented in an enterprise network.

Understanding the Function of DHCP

This topic explains the purpose and function of DHCP.



Dynamic Host Configuration Protocol

DHCP client broadcasts discover message.

DHCP server unicasts offer message.

DHCP client responds with request message.

DHCP server completes the process with acknowledgment message.

BSCI v3.0—5-3

A Cisco IOS device can be a DHCP server, a DHCP client, or a DHCP relay agent.

The figure shows the steps that occur when a DHCP client requests an IP address from a DHCP server.

1. The host sends a DHCPDISCOVER broadcast message to locate a DHCP server.

2. A DHCP server offers configuration parameters such as an IP address, a MAC address, a domain name, a default gateway, and a lease for the IP address to the client in a DHCPOFFER unicast message. Also included may be IP telephony DHCP options such as option 150, which is used for TFTP configuration of IP telephones.

3. The client returns a formal request for the offered IP address to the DHCP server in a DHCPREQUEST broadcast message.

4. The DHCP server confirms that the IP address has been allocated to the client by returning a DHCPACK unicast message to the client.

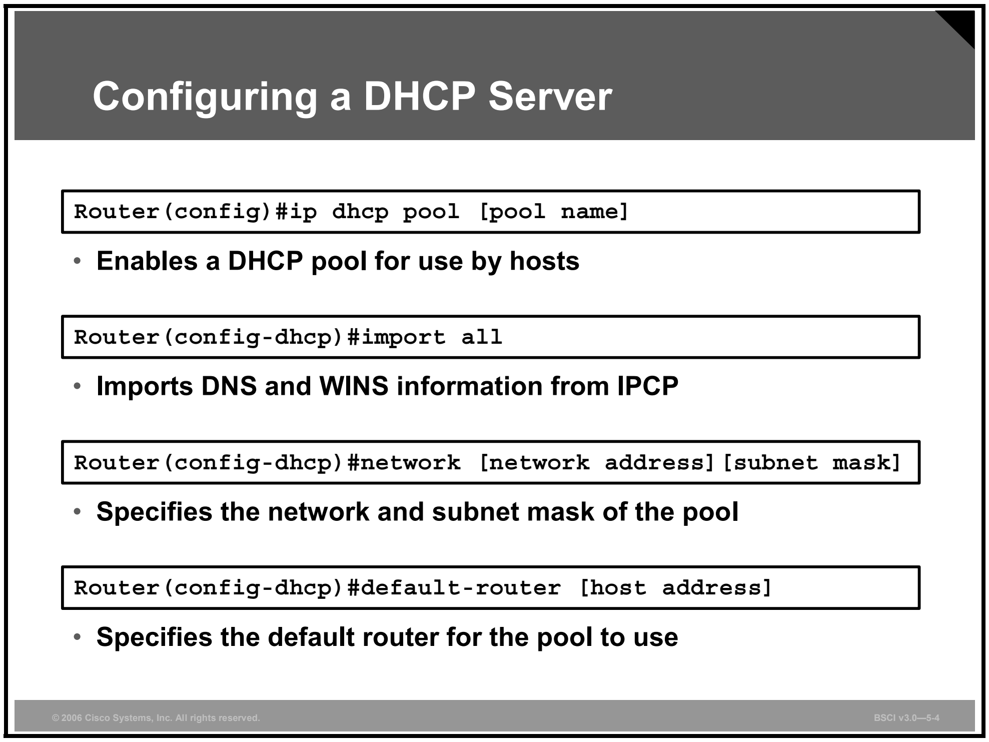
A DHCP client may receive offers from multiple DHCP servers and can accept any one of the offers. However, the client usually accepts the first offer that it receives. Also, the offer from the DHCP server is not a guarantee that the IP address will be allocated to the client. The server usually reserves the address until the client has had a chance to formally accept the address.

DHCP supports three possible address allocation mechanisms:

- **Manual:** The network administrator assigns the IP address to a specific MAC address. DHCP is used to dispatch the assigned address to the host.
- **Automatic:** The IP address is permanently assigned to a host.
- **Dynamic:** The IP address is assigned to a host for a limited time or until the host explicitly releases the address. This mechanism supports automatic address reuse when the host to which the address has been assigned no longer needs the address.

Configuring DHCP

This topic explains how to configure DHCP on Cisco IOS software.



Configuring a DHCP Server

```
Router(config)#ip dhcp pool [pool name]
```

- **Enables a DHCP pool for use by hosts**

```
Router(config-dhcp)#import all
```

- **Imports DNS and WINS information from IPCP**

```
Router(config-dhcp)#network [network address][subnet mask]
```

- **Specifies the network and subnet mask of the pool**

```
Router(config-dhcp)#default-router [host address]
```

- **Specifies the default router for the pool to use**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-4

The following are tasks for configuring a Cisco IOS DHCP server:

- Configuring a DHCP database agent or disabling DHCP conflict logging
- Configuring a DHCP address pool (required)
- Excluding IP addresses
- Enabling the Cisco IOS DHCP server and relay agent features
- Configuring manual bindings
- Configuring a DHCP server boot file
- Configuring the number of ping packets and timeout
- Enabling the Cisco IOS DCHP client on Ethernet interfaces
- Performing a DHCP server options import and autoconfiguration
- Configuring the relay agent information option in BOOTREPLY messages
- Configuring a relay agent information reforwarding policy
- Enabling the DHCP smart-relay feature

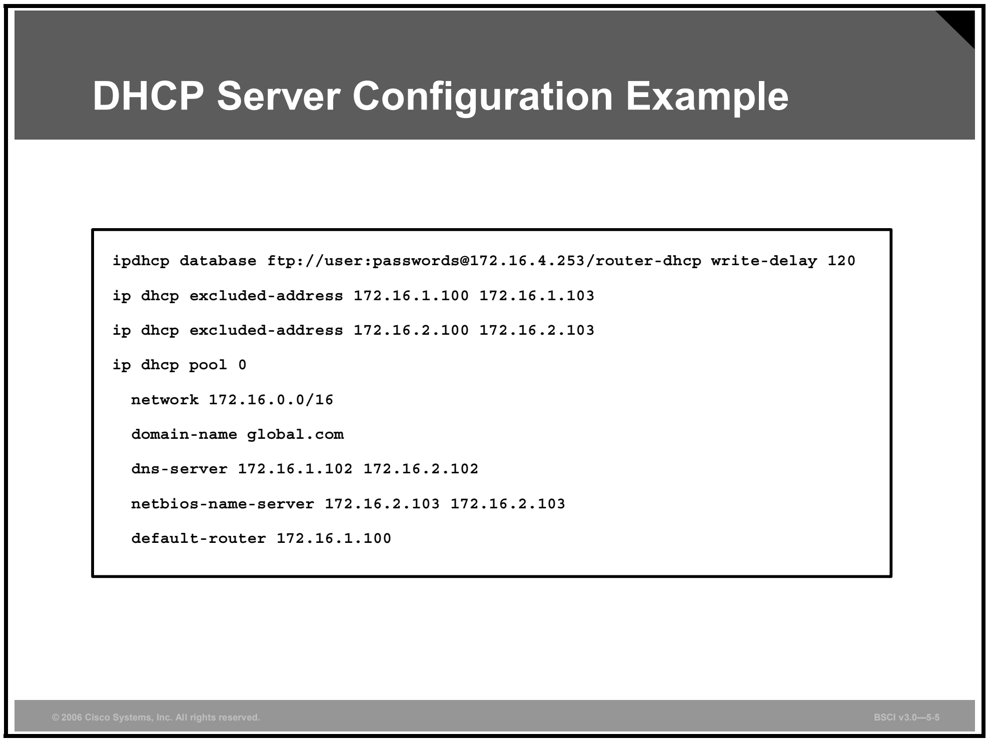
Not all of the optional features are covered in detail. Additional information on these optional features is available on Cisco.com.

Cisco IOS DHCP Server Commands and Parameters

This table identifies the major commands to implement Cisco IOS DHCP server and options.

Command and Parameter	Description
Router(config)#**service dhcp**	Enables DHCP features on router; it is on by default.
Router(config)#**ip dhcp database** *url* [**timeout** seconds | *write-delay seconds*]	Configures the database agent and the interval between database updates and database transfers.
Router(config)#**no ip dhcp conflict logging**	Disables DHCP conflict logging. (Used if a DHCP database agent is not configured.)
Router(config)#**ip dhcp excluded-address** *low-address* [*high address*]	Specifies the IP address that the DHCP server should not assign to DHCP clients.
Router(config)#**ip dhcp pool** *name*	Creates a name for the DCHP server address pool and places you in DHCP pool configuration mode.
Router(config-dhcp)#**network** *network-number* [*mask* | *prefix-length*]	Specifies the subnet network number and mask of the DHCP address pool.
Router(config-dhcp)#**domain-name** *domain*	Specifies the domain name for the client.
Router(config-dhcp)#**dns-server** *address* [*address2...address8*]	Specifies the IP address of a Domain Name System (DNS) server that is available to a DHCP client. One is required, but up to eight can be specified.
Router(config-dhcp)#**netbios-name-server** *address* [*address2...address8*]	Same as DNS, but for Windows Internet Name Service (WINS).
Router(config-dhcp)#**default-router** *address* [*address2... address8*]	Specifies the IP address of the default router for a DHCP client.
Router(config-dhcp)#**lease** {*days* [*hours*] [*minutes*] | *infinite*}	Specifies the duration of the lease. The default is a one-day lease.
Router(config-dhcp)#**import all**	Used to import DHCP option parameters into the DHCP server database. Used for remote DHCP pools.

Additional commands are available to customize manual bindings for individual clients, including MAC addresses. Additional options are also available with implementation of the DHCP relay agent function.



DHCP Server Configuration Example

```
ipdhcp database ftp://user:passwords@172.16.4.253/router-dhcp write-delay 120
ip dhcp excluded-address 172.16.1.100 172.16.1.103
ip dhcp excluded-address 172.16.2.100 172.16.2.103
ip dhcp pool 0
  network 172.16.0.0/16
  domain-name global.com
  dns-server 172.16.1.102 172.16.2.102
  netbios-name-server 172.16.2.103 172.16.2.103
  default-router 172.16.1.100
```

Before configuring the Cisco IOS DHCP Server feature, you should complete the following tasks:

1. Identify an external FTP, TFTP, or Remote Copy Protocol (RCP) server that will be used to store the DHCP binding database.

2. Identify the IP address range to be assigned by the DHCP server and the IP addresses to be excluded (default routers and other statically assigned addresses within a dynamically assigned range).

3. Identify DHCP options for devices where necessary, including these:

   — Default boot image name

   — Default routers

   — DNS servers

   — Network BIOS (NetBIOS) name server

   — IP telephony options, such as option 150

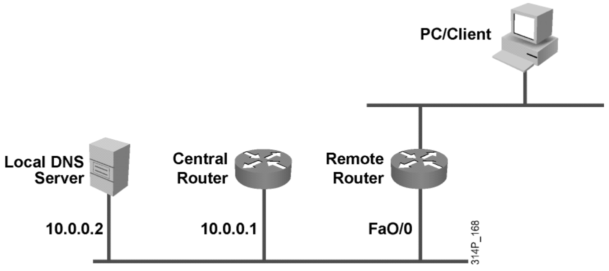
4. Decide on a DNS domain name.

Cisco IOS devices can also be configured as DHCP clients and DHCP relay agents.

In the following configuration example, two DHCP address pools are created, one for subnet 172.16.1.0 and one for subnet 172.16.2.0. The host will receive its address from the DHCP server on the closest device. The router determines how to provide the addresses based on the IP address on the router interface.

```
!
ip dhcp pool 1
  network 172.16.1.0/24
  domain-name global.com
  dns-server 172.16.1.102
  netbios-name-server 172.16.1.103
  default-router 172.16.1.100 172.16.1.101
  lease 30
!
ip dhcp pool 2
  network 172.16.2.0/24
  domain-name global.com
  dns-server 172.16.2.102
  netbios-name-server 172.16.2.103
  default-router 172.16.2.100 172.16.2.101
  lease 30
```

Importing and Autoconfiguration



BSCI v3.0—5-6

In the past, Cisco IOS DHCP servers had to be configured on each device separately with all parameters and options specified in each case. Cisco IOS software has been revised to allow remote Cisco IOS DHCP servers to be configured to import option parameters from a centralized server.

The following is an example of the partial command syntax for this feature:

Central Router

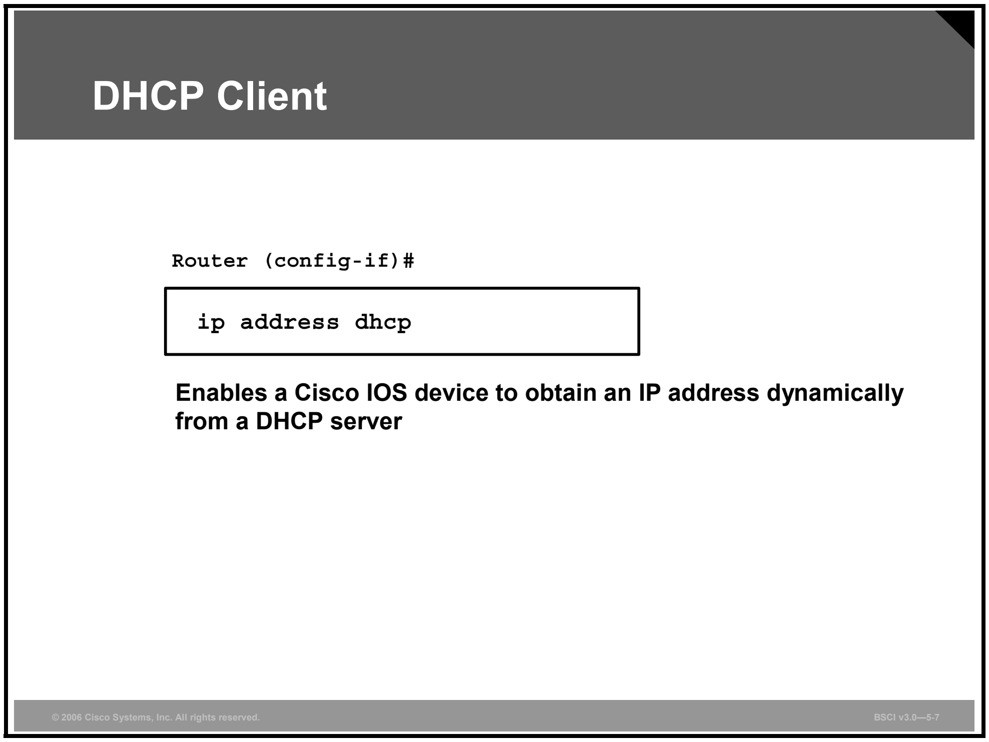
```
 ip dhcp-excluded address 10.0.0.1 10.0.0.5
 ip dhcp pool central
 network 10.0.0.0 255.255.255.0
 default-router 10.0.0.1 10.0.0.5
 domain name central.com
 dns-server 10.0.0.2
 netbios-name-server 10.0.0.2
 interface fastethernet0/0
   ip address 10.0.0.1 255.255.255.0
```

Remote Router

```
 ip dhcp pool client
 network 20.0.0.0 255.255.255.0
 ip dhcp-excluded address 20.0.0.2
 default-router 20.0.0.2
 import all
interface fastethernet0/0
  ip address dhcp
```

Configuring the DHCP Client

This topic explains how to configure the DHCP client.



DHCP Client

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

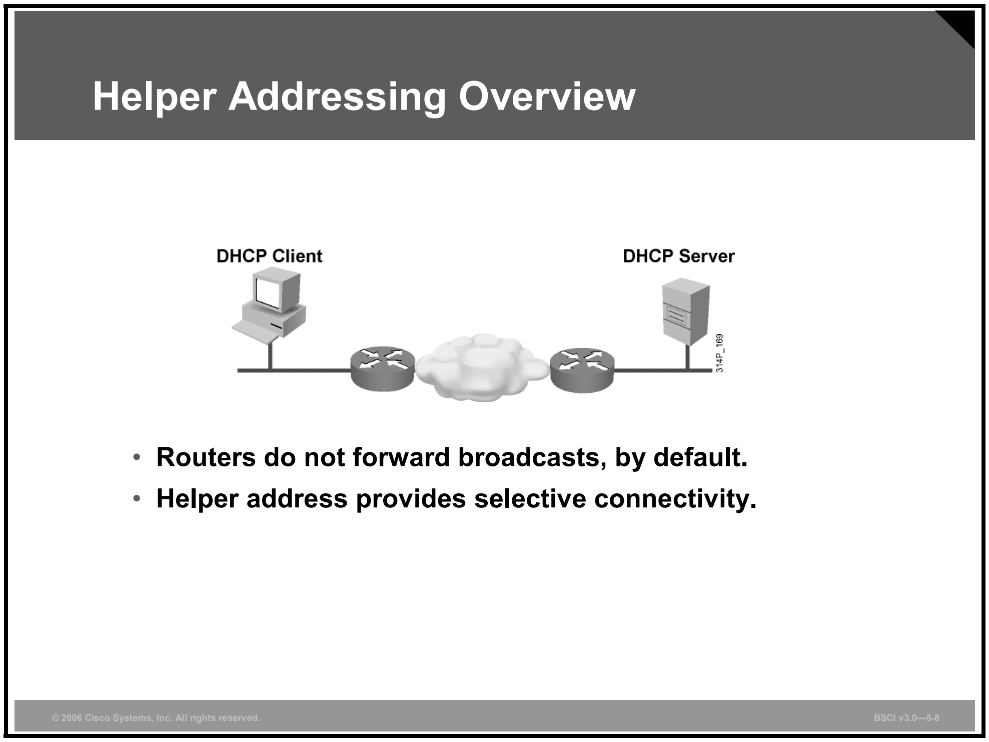
```
ip address dhcp
```

Enables a Cisco IOS device to obtain an IP address dynamically from a DHCP server

A Cisco IOS device can be configured to be a DHCP client and obtain an interface address dynamically from a DHCP server with the command **ip address dhcp**. This command is implemented in interface mode and is specific to an individual interface.

Explaining the IP Helper Address

This topic explains the purpose and configuration of IP helper addresses.



Helper Addressing Overview

DHCP Client
DHCP Server
314P_169

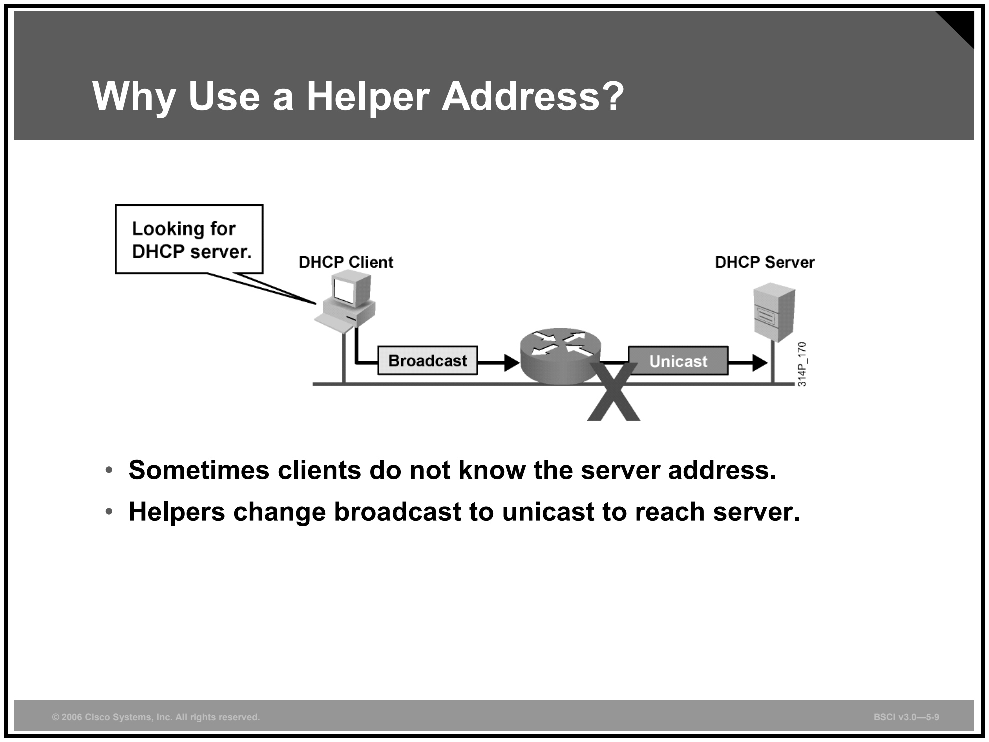
- **Routers do not forward broadcasts, by default.**
- **Helper address provides selective connectivity.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-8

The Cisco IOS DHCP relay agent is enabled on an interface only when the **ip helper-address** is configured.

The DHCP relay agent is any host that forwards DHCP packets between clients and servers. Relay agents are used to forward requests and replies between clients and servers when they are not on the same physical subnet.

Relay agents receive DHCP messages and then generate a new DHCP message to send out on another interface.



Why Use a Helper Address?

- Sometimes clients do not know the server address.
- Helpers change broadcast to unicast to reach server.

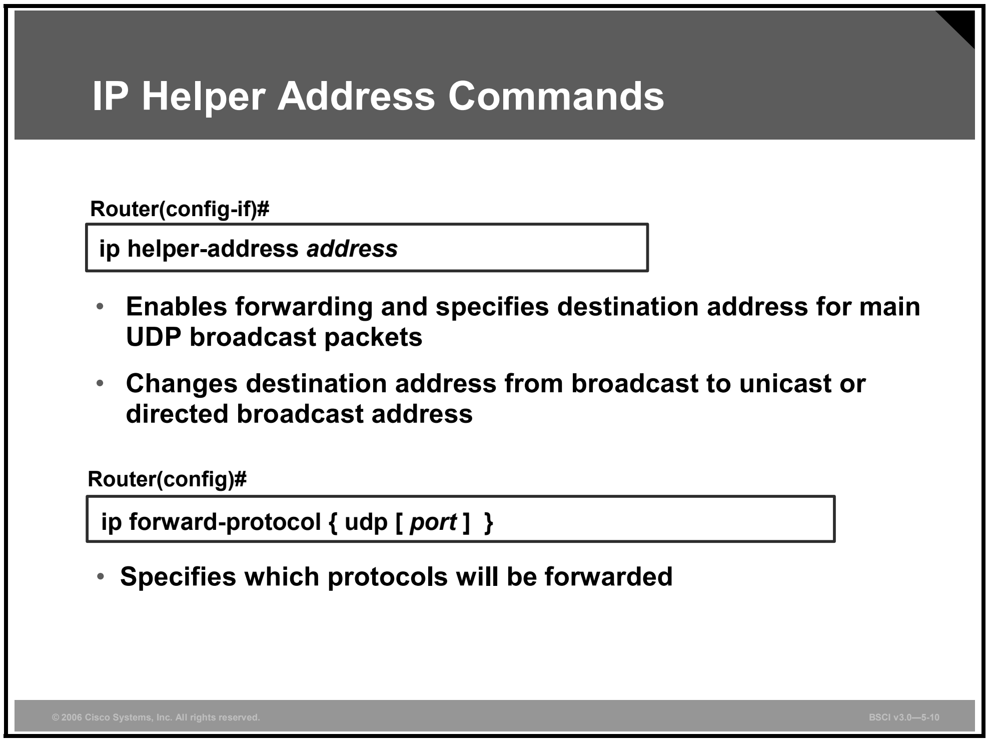
BSCI v3.0—5-9

DHCP clients use UDP broadcasts to send their initial DHCPDISCOVER message, because they do not have information about the network to which they are attached.

If the client is on a network that does not include a server, UDP broadcasts are normally not forwarded by the attached router.

The **ip helper-address** command causes the UDP broadcast to be changed to a unicast and forwarded out another interface to a unicast IP address specified by the command.

The relay agent sets the gateway address (giaddr field of the DHCP packet) and, if configured, adds the relay agent information option (option 82) in the packet and forwards it to the DHCP server. The reply from the server is forwarded back to the client after removing option 82.



IP Helper Address Commands

Router(config-if)#

```
ip helper-address address
```

- **Enables forwarding and specifies destination address for main UDP broadcast packets**
- **Changes destination address from broadcast to unicast or directed broadcast address**

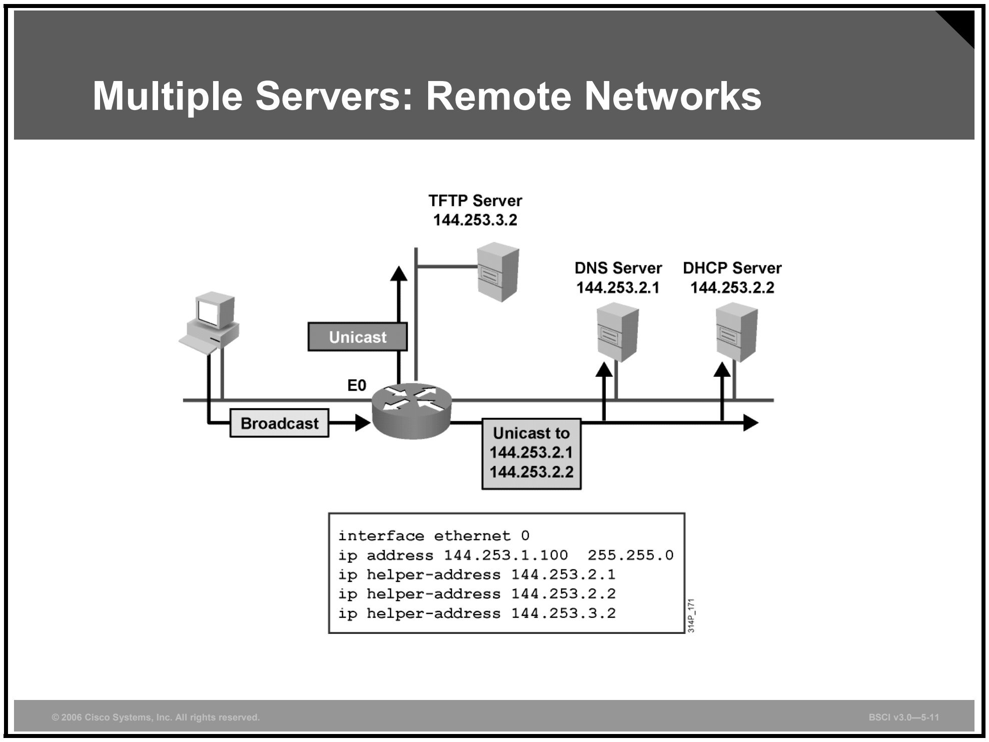
Router(config)#

```
ip forward-protocol { udp [ port ] }
```

- **Specifies which protocols will be forwarded**

The figure illustrates the use of the **ip helper-address** command to implement DHCP relay agent.

However, the command enables forwarding of all of the well-known UDP ports that may be included in an UDP broadcast message. The **ip forward-protocol udp** command can be used to customize this feature to network requirements.



Multiple Servers: Remote Networks

```
interface ethernet 0
ip address 144.253.1.100  255.255.0
ip helper-address 144.253.2.1
ip helper-address 144.253.2.2
ip helper-address 144.253.3.2
```

BSCI v3.0—5-11

The UDP well-known ports identified by default forwarding UDP services are these:

- **Time:** 37
- **TACACS:** 49
- **DNS:** 53
- **BOOTP/DHCP server:** 67
- **BOOTP/DHCP client:** 68
- **TFTP:** 69
- **NetBIOS name service:** 137
- **NetBIOS datagram service:** 138

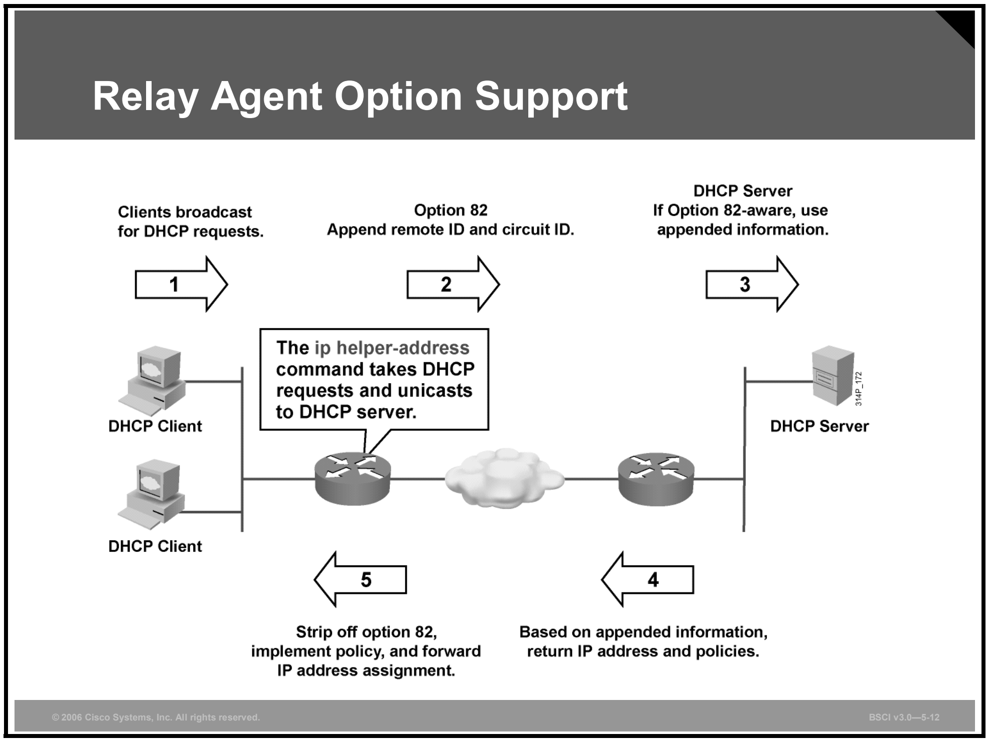
Ports can be eliminated from the forwarding service with the **no ip forward-protocol** command, and ports can be added to the forwarding service with the **ip forward-protocol** command. The following is an example:

```
!
interface fastethernet0/0
ip address 10.3.3.3 255.255.255.0
ip helper-address 10.1.1.1
no ip forward-protocol udp 137
no ip forward-protocol udp 138
no ip forward-protocol udp 37
ip forward-protocol udp 8000
```

This configuration would cause time and NetBIOS ports to not be forwarded, and UDP port 8000 would be added to the forwarded list.

Configuring DHCP Relay Services

This topic explains the purpose and configuration of DHCP relay services.



Relay Agent Option Support

Clients broadcast for DHCP requests.

Option 82
Append remote ID and circuit ID.

DHCP Server
If Option 82-aware, use appended information.

The ip helper-address command takes DHCP requests and unicasts to DHCP server.

Based on appended information, return IP address and policies.

Strip off option 82, implement policy, and forward IP address assignment.

BSCI v3.0—5-12

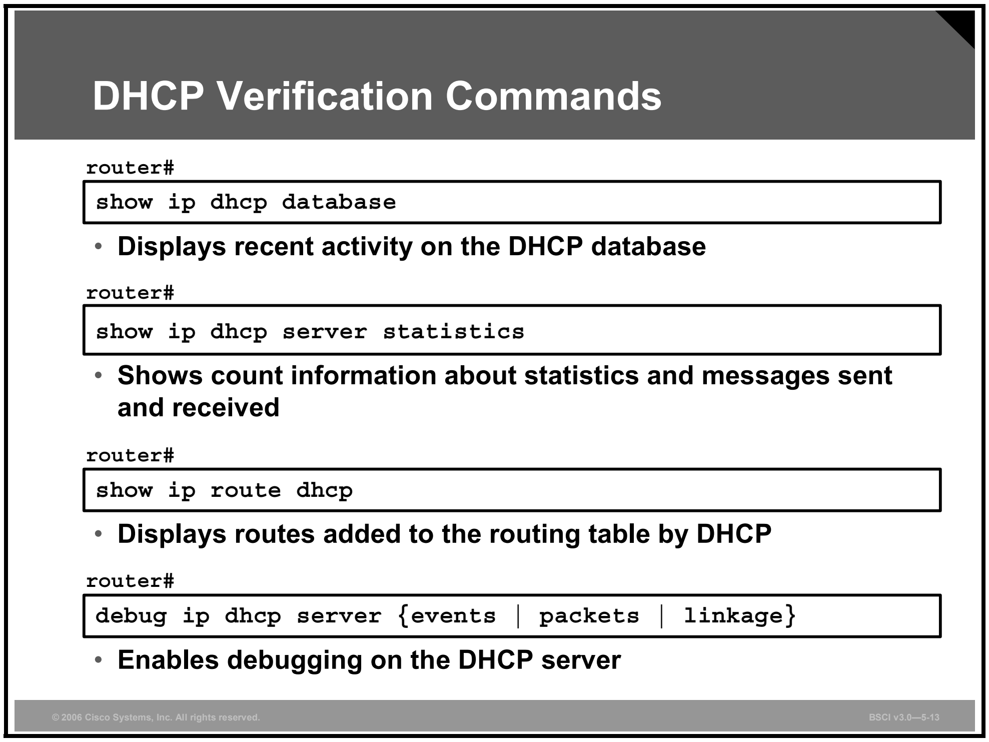
When you use the **ip dhcp relay information option** command, the relay agent adds the circuit identifier suboption and the remote ID suboption to the relay agent information and forwards them to a DHCP server. The following explains the DHCP relay services process:

1. The DHCP client generates a DHCP request and broadcasts it on the network.
2. The DHCP relay agent intercepts the broadcast DHCP request packet and inserts the relay agent information option (82) in the packet. The relay agent option contains the related suboptions.
3. The DHCP relay agent unicasts the DHCP packet to the DHCP server.
4. The DHCP server receives the packet and uses the suboptions to assign IP addresses and other configuration parameters and forwards them back to the client.
5. The suboption fields are stripped off of the packet by the relay agent while forwarding to the client.

Description of Option Support Commands and Parameters

The table lists the option support commands.

Command and Parameter	Description
`Router(config)#`**`ip dhcp information option`**	Enables the system to insert the DHCP relay agent information option (82) in forwarded BOOTREQUEST messages to a DHCP server. Disabled by default.
`Router(config)#`**`ip dhcp information check`**	Configures DHCP to check that the relay agent information option in forwarded BOOTREPLY messages is valid. This command is used only to re-enable this function if it is disabled; it is on by default.
`Router(config)#`**`ip dhcp information policy`** {*drop* | *keep* | *replace*}	Configures the reforwarding policy for the relay agent (specifies what a relay agent should do if a message already contains relay information).
`Router(config)#`**`ip dhcp relay information trust-all`**	Configures all information on a router as trusted sources of the DHCP relay information option. Useful when Ethernet switches are involved in delivery of packet. For an individual interface, the **ip dhcp relay information trusted** command can be used.
`Router#`**`show ip dhcp-relay information trusted-sources`**	Displays all interfaces configured to be a trusted source for the DHCP relay information option.



DHCP Verification Commands

```
router#
show ip dhcp database
```

- **Displays recent activity on the DHCP database**

```
router#
show ip dhcp server statistics
```

- **Shows count information about statistics and messages sent and received**

```
router#
show ip route dhcp
```

- **Displays routes added to the routing table by DHCP**

```
router#
debug ip dhcp server {events | packets | linkage}
```

- **Enables debugging on the DHCP server**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-13

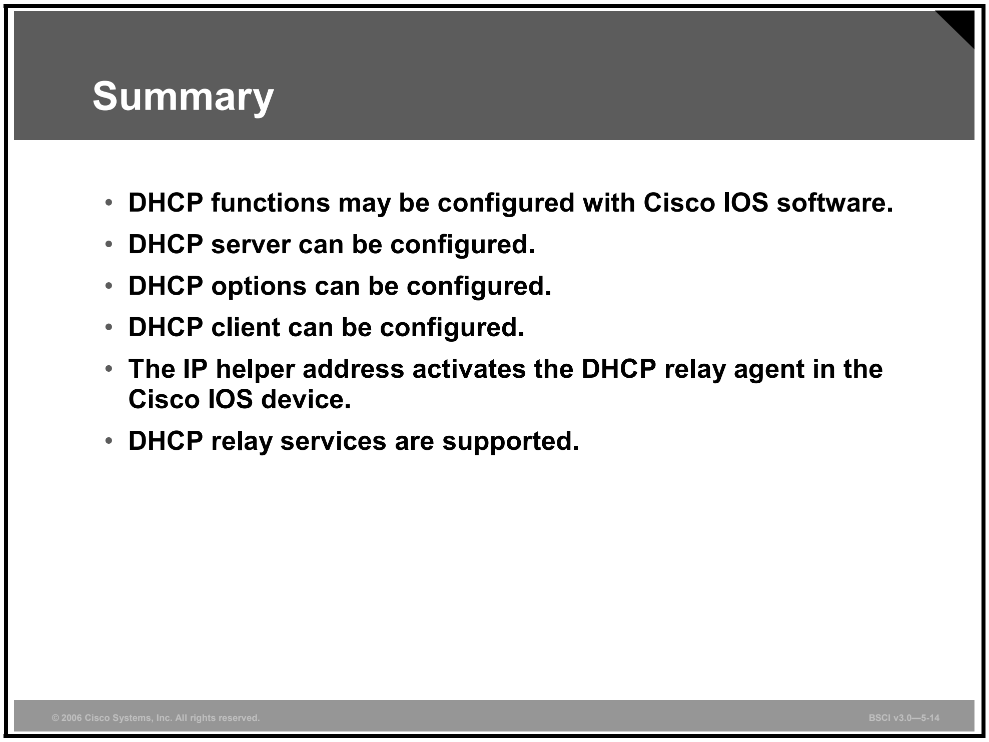
Description of DHCP Verification Commands and Parameters

The table provides a list of useful DHCP verification commands.

Command and Parameter	Description
show ip dhcp database	Displays recent activity in the DHCP database
show ip dhcp server statistics	Displays count information about server statistics and messages sent and received
show ip route dhcp {*ip-address*}	Displays the routes added to the routing table by the Cisco IOS DHCP server and relay agent
show ip dhcp binding {*address*}	Displays a list of all bindings created on a specific DHCP server
show ip dhcp conflict	Displays a list of all address conflicts recorded by a specific DHCP server
show ip dhcp import	Displays the option parameters that were imported into the DHCP server database
clear ip dhcp binding {*address* | *}	Deletes an automatic address binding from the DHCP database
clear ip dhcp conflict {*address* | *}	Clears an address conflict from the DHCP database
clear ip dhcp server statistics	Resets all DHCP server counters to 0
clear ip route dhcp {*ip-address*}	Removes routes from the routing table added by the Cisco IOS DHCP server and relay agent

Summary

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



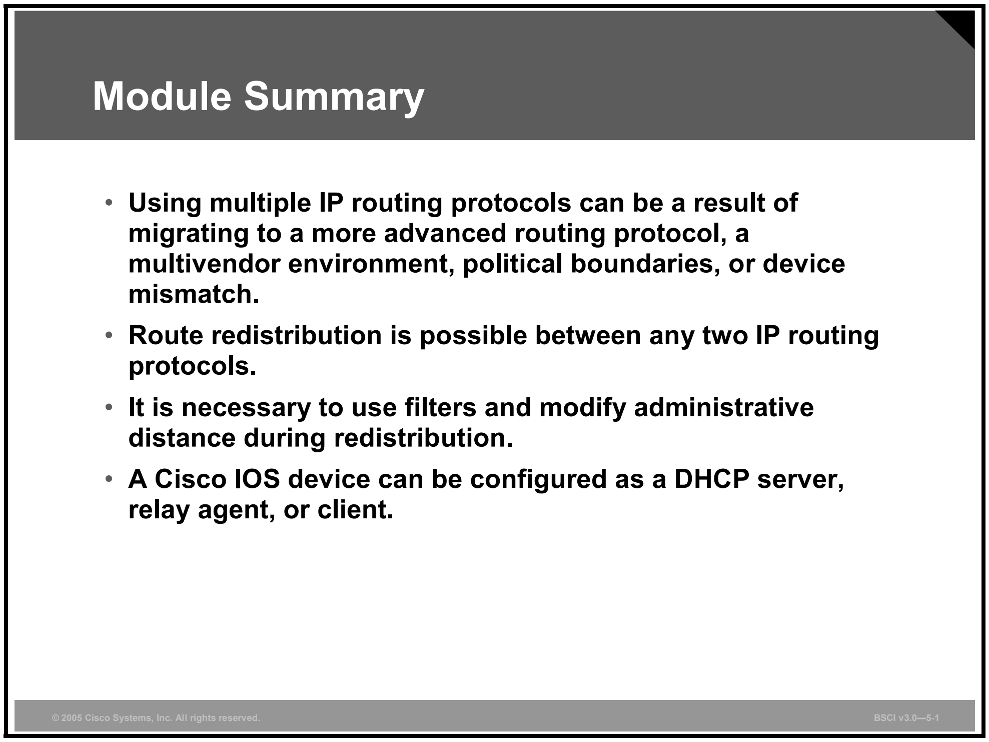
Summary

- DHCP functions may be configured with Cisco IOS software.
- DHCP server can be configured.
- DHCP options can be configured.
- DHCP client can be configured.
- The IP helper address activates the DHCP relay agent in the Cisco IOS device.
- DHCP relay services are supported.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-14

Module Summary

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



Module Summary

- **Using multiple IP routing protocols can be a result of migrating to a more advanced routing protocol, a multivendor environment, political boundaries, or device mismatch.**
- **Route redistribution is possible between any two IP routing protocols.**
- **It is necessary to use filters and modify administrative distance during redistribution.**
- **A Cisco IOS device can be configured as a DHCP server, relay agent, or client.**

© 2005 Cisco Systems, Inc. All rights reserved. BSCI v3.0—5-1

This module covered IP route redistribution and the control of redistributed routing updates. It also covered the use of passive interfaces and route maps for this control. The use of route maps for policy-based routing (PBR) was also covered. Finally, using a Cisco IOS device such as a DHCP server, relay agent, or client was described.

Any two IP routing protocols can be redistributed. However, many types of incorrect information may be propagated. Passive interfaces, distribute lists, and route maps are some of the methods used to control these updates. Route maps may also be used to implement PBR for cost savings, quality of service (QoS), and other purposes driven by enterprise policy.

An advanced Cisco IOS feature is the use of a Cisco IOS device as a DHCP server, DHCP relay agent, or DHCP client. The **ip helper-address** command triggers the use of a Cisco IOS device as a relay agent, and numerous additional options can be implemented.

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 situations might require multiple routing protocols in a network? (Choose three.) (Source: Operating a Network Using Multiple IP Routing Protocols)

A) when a new Layer 2-only switch is added to the network
B) when you are migrating from one routing protocol to another
C) when you are using routers from multiple vendors
D) when there are host-based routers from multiple vendors

Q2) Which two routing protocols can be redistributed into OSPF by a Cisco router? (Choose two.) (Source: Configuring and Verifying Route Redistribution)

A) IP EIGRP
B) Appletalk EIGRP
C) RIPv2
D) IPX RIP

Q3) Which is a reason for avoiding doing route redistribution on two routers between the same two routing domains? (Source: Configuring and Verifying Route Redistribution)

A) higher cost of two routers
B) routing feedback
C) Cisco IOS incompatibility
D) not possible to use two routers

Q4) The **subnet** keyword is required when you are redistributing subnet routes into which routing protocol? (Source: Configuring and Verifying Route Redistribution)

A) OSPF
B) RIP
C) EIGRP
D) IS-IS

Q5) Which two actions does issuing the **passive-interface** command prevent? (Choose two.) (Source: Controlling Routing Update Traffic)

A) prevents routing updates from being sent out an interface but not from being received on an interface
B) prevents routing updates from being sent out an interface and also from being received on an interface
C) prevents link-state protocols and EIGRP from sending hellos out the interface
D) prevents the exchange of routing updates only, not hellos

Q6) Which statement does not describe route map operation? (Source: Controlling Routing Update Traffic)

A) Routes maps use a top-down processing scheme.
B) Route maps use match and set logic to match a metric and then set which route is redistributed.
C) Route maps are line-numbered for easier editing.
D) Route maps use an implicit deny at the bottom of the map, just like access lists.

Q7) When you are configuring a route map, you must define a map tag. How is a map tag used? (Source: Controlling Routing Update Traffic)

A) to tag a route in a **set** command
B) to match on a tagged route in a **match** command
C) to deny routes from being redistributed
D) to give a name to the route map

Q8) What does administrative distance rank? (Source: Controlling Routing Update Traffic)

A) metrics
B) sources of routing information
C) router reliability
D) best paths

Q9) DHCP is structured to use which two well-known UDP ports? (Choose two) (Source: Implementing Advanced Cisco IOS Features: Configuring DHCP)

A) BOOTPS
B) TFTP
C) DNS
D) BOOTP

Q10) A request for address assignment is initiated by which device? (Source: Implementing Advanced Cisco IOS Features: Configuring DHCP)

A) client
B) relay agent
C) router
D) DHCP server

Q11) What is the purpose of the Cisco IOS global command **ip dhcp pool** [**pool name**]**?** (Source: Implementing Advanced Cisco IOS Features: Configuring DHCP)

A) to define the inside global address pool for NAT
B) to define a range of IP addresses to be assigned by the router acting as a DHCP server
C) to define exceptions to the IP addresses to be assigned
D) to enable global route summarization

Module Self-Check Answer Key

Q1)	B, C, D
Q2)	A, C
Q3)	B
Q4)	A
Q5)	A, C
Q6)	B
Q7)	D
Q8)	B
Q9)	A, D
Q10)	A
Q11)	B

Module 6

Implementing BGP

Overview

The Internet has become a vital resource in many organizations, resulting in redundant connections to multiple Internet service providers (ISPs). With multiple connections, Border Gateway Protocol (BGP) is an alternative to using default routes to control path selections.

A BGP administrator must understand the various options involved in properly configuring BGP for scalable internetworking. This module discusses BGP configuration and verification for enterprise ISP connectivity.

Module Objectives

Upon completing this module, you will be able to implement and verify BGP for enterprise ISP connectivity. This ability includes being able to meet these objectives:

- Explain BGP concepts and terminology
- Describe the structural concepts of a BGP network
- Implement BGP operation
- Manipulate BGP path selection using BGP attributes
- Manipulate BGP traffic using route maps

Lesson 1

Explaining BGP Concepts and Terminology

Overview

This lesson introduces Border Gateway Protocol (BGP) as an exterior gateway protocol (EGP) and describes how BGP may be used by enterprises to connect to Internet service providers (ISPs). BGP routes between autonomous systems, which are defined in this lesson as they relate to BGP and interior gateway protocols (IGPs).

This lesson also examines various aspects of BGP, such as its nature as a path-vector routing protocol and how that functionality allows routing policy decisions at the autonomous system (AS) level to be enforced. Other characteristics, such as how BGP uses TCP as its transport function, are defined, along with the four message types that BGP uses to communicate with other BGP routers.

This lesson introduces the key concepts of BGP and identifies the differences between BGP and IGPs, such as the Open Shortest Path First Protocol (OSPF), Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Enhanced IGRP (EIGRP), and Intermediate System-to-Intermediate System Protocol (IS-IS).

Understanding the important characteristics of BGP and the way in which it behaves differently from IGPs is necessary for knowing when and when not to use BGP.

Objectives

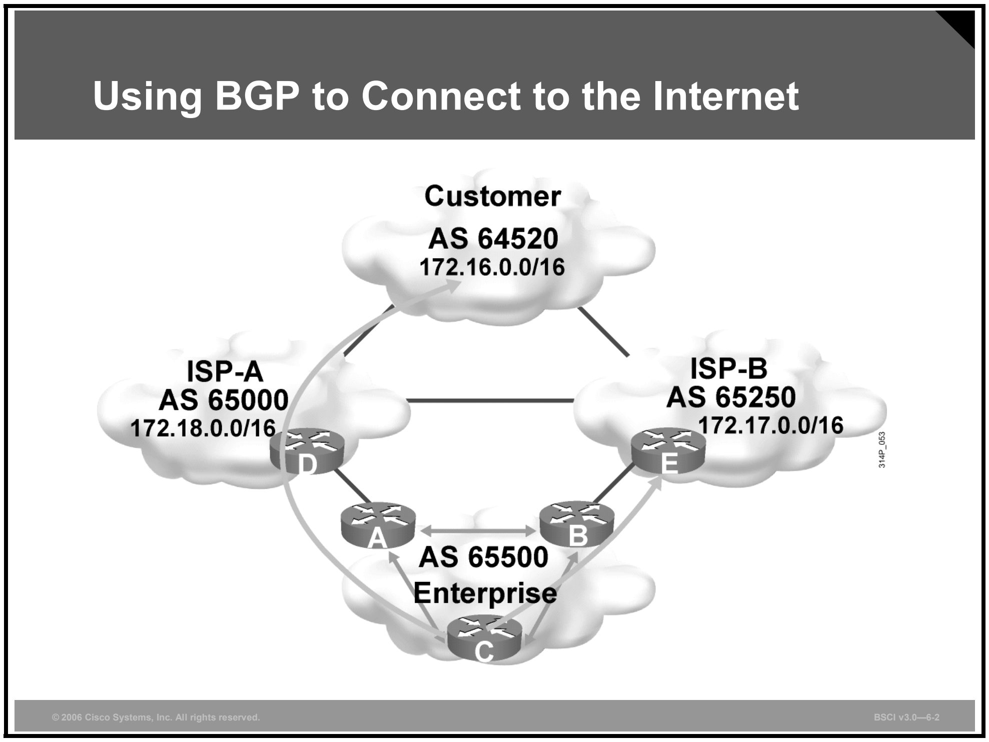
Upon completing this lesson, you will be able to explain BGP concepts and terminology. This ability includes being able to meet these objectives:

- Describe connectivity between an enterprise network and an ISP that requires the use of BGP, including a description of the issues that arise when an enterprise decides to connect to the Internet through multiple ISPs
- Describe BGP multihoming options
- Describe how BGP routes between autonomous systems
- Describe how BGP uses path-vector functionality

- Describe the features of BGP in terms of deployment, enhancements over other distance vector routing protocol and database types
- Describe the functionality of each BGP message type

Using BGP in an Enterprise Network

This topic describes connectivity between an enterprise network and an ISP that requires the use of BGP, including a description of the issues that arise when an enterprise decides to connect to the Internet through multiple ISPs.



Using BGP to Connect to the Internet

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-2

The Internet is a collection of autonomous systems that are interconnected to allow communication among them. BGP provides the routing between these autonomous systems.

Enterprises that want to connect to the Internet do so through one or more ISPs. If your organization has only one connection to one ISP, then you probably do not need to use BGP; instead you would use a default route. However, if you have multiple connections to one or to multiple ISPs, then BGP might be appropriate because it allows manipulation of path attributes, so that the optimal path can be selected.

To understand BGP, you first need to understand the way in which it is different from the other protocols discussed so far in this course. One way to categorize routing protocols is by whether they are interior or exterior:

- **IGP:** A routing protocol that exchanges routing information within an AS. RIP, IGRP, OSPF, IS-IS, and EIGRP are examples of IGPs.
- **EGP:** A routing protocol that exchanges routing information between different autonomous systems. BGP is an example of an EGP.

BGP is an interdomain routing protocol (IDRP), also known as an EGP. BGP version 4 (BGP4) is the latest version of BGP and is defined in RFC 4271. As noted in this RFC, the classic definition of an AS is “a set of routers under a single technical administration, using an IGP and common metrics to route packets within the autonomous system, and using an inter-autonomous system routing protocol (also called an EGP) to determine how to route packets to other autonomous systems.”

Autonomous systems can use more than one IGP, potentially with several sets of metrics. From the BGP point of view, the most important characteristic of an AS is that it appears to other autonomous systems to have a single coherent interior routing plan and presents a consistent picture of reachable destinations. All parts of an AS must connect to each other.

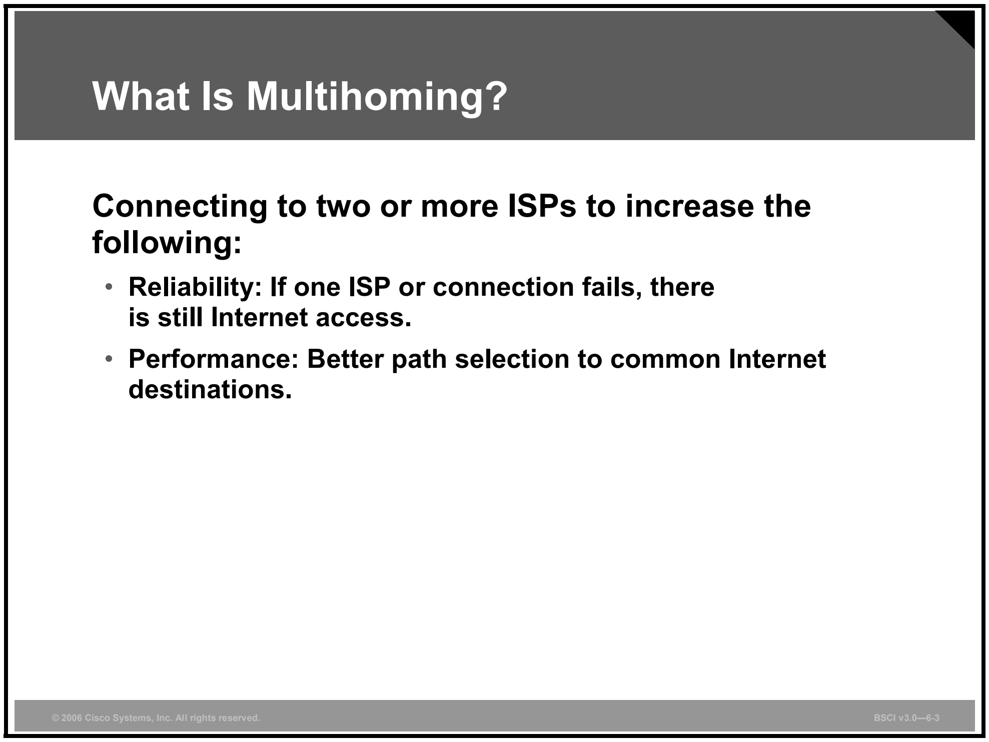
When BGP is running between routers in different autonomous systems, it is called External BGP (EBGP). When BGP is running between routers in the same AS, it is called Internal BGP (IBGP). BGP allows the path that packets take to be manipulated by the AS, as described in this module. It is important to understand how BGP works to avoid creating problems for your AS as a result of running BGP.

For example, enterprise AS 65500 in the figure is learning routes from both ISP-A and ISP-B via EBGP and is also running IBGP on all of its routers. AS 65500 learns about routes and chooses the best way to each one based on the configuration of the routers in the AS and the BGP routes passed from the ISPs. If one of the connections to the ISPs goes down, traffic will be sent through the other ISP.

One of the routes that AS 65500 learns from ISP-A is the route to 172.18.0.0/16. If that route is passed through AS 65500 using IBGP and is mistakenly announced to ISP-B, then ISP-B may decide that the best way to get to 172.18.0.0/16 is through AS 65500, instead of through the Internet. AS 65500 would then be considered a transit AS, which is a very undesirable situation. AS 65500 wants to have a redundant Internet connection, but does not want to act as a transit AS between the two ISPs. Careful BGP configuration is required to avoid this situation.

BGP Multihoming Options

This topic describes BGP multihoming options.



What Is Multihoming?

Connecting to two or more ISPs to increase the following:

- **Reliability: If one ISP or connection fails, there is still Internet access.**
- **Performance: Better path selection to common Internet destinations.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-3

Multihoming is when an AS has more than one connection to the Internet. Two typical reasons for multihoming are as follows:

- **To increase the reliability of the connection to the Internet:** If one connection fails, the other connection remains available.
- **To increase the performance of the connection:** Better paths can be used to certain destinations.

The benefits of BGP are apparent when an AS has multiple EBGP connections to either a single AS or multiple autonomous systems. Having multiple connections allows an organization to have redundant connections to the Internet so that if a single path becomes unavailable, connectivity can still be maintained.

An organization can be multihomed to either a single ISP or to multiple ISPs. A drawback to having all of your connections to a single ISP is that connectivity issues in that single ISP can cause your AS to lose connectivity to the Internet. By having connections to multiple ISPs, an organization gains the following benefits:

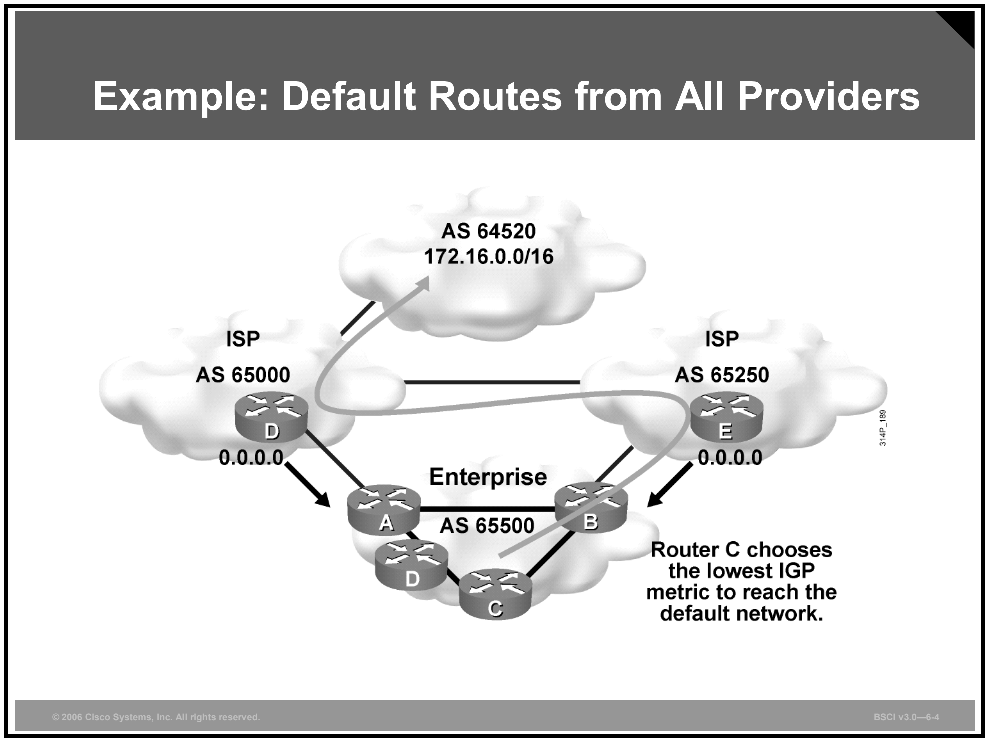
- Has redundancy with the multiple connections
- Is not tied into the routing policy of a single ISP
- Has more paths to the same networks for better policy manipulation

A multihomed AS will run EBGP with its external neighbors and might also run IBGP internally.

If an organization has determined that it will perform multihoming with BGP, there are three common ways to do this:

- **Each ISP passes only a default route to the AS:** The default route is passed to the internal routers.
- **Each ISP passes only a default route and provider-owned specific routes to the AS:** These routes may be passed to internal routers, or all internal routers in the transit path can run BGP and pass these routes between them.
- **Each ISP passes all routes to the AS:** All internal routers in the transit path run BGP and pass these routes between them.

These options are described in the following topics.



Example: Default Routes from All Providers

Router C chooses the lowest IGP metric to reach the default network.

BSCI v3.0—6-4

The first multihoming option is to receive only a default route from each ISP. This configuration requires the fewest resources within the AS because a default route is used to reach any external destinations. The AS sends all its routes to the ISPs, which process and pass them on to other autonomous systems.

If a router in the AS learns about multiple default routes, the local interior routing protocol installs the best default route in the routing table. From the perspective of this router, it takes the default route with the least-cost IGP metric. This IGP default route routes packets destined to the external networks to an edge router of this AS, which is running EBGP with the ISPs. The edge router uses the BGP default route to reach all external networks.

The route that inbound packets take to reach the AS is decided outside the AS (within the ISPs and other autonomous systems).

Regional ISPs that have multiple connections to national or international ISPs commonly implement this option. The regional ISPs do not use BGP for path manipulation; however, they require the ability to add new customers as well as the networks of the customers. If the regional ISP does not use BGP, then each time that the regional ISP adds a new set of networks, the customers must wait until the national ISPs add these networks to their BGP process and place static routes pointing at the regional ISP. By running EBGP with the national or international ISPs, the regional ISP needs to add only the new networks of the customers to its BGP process. These new networks automatically propagate across the Internet with minimal delay.

A customer that chooses to receive default routes from all providers must understand the limitations of this option:

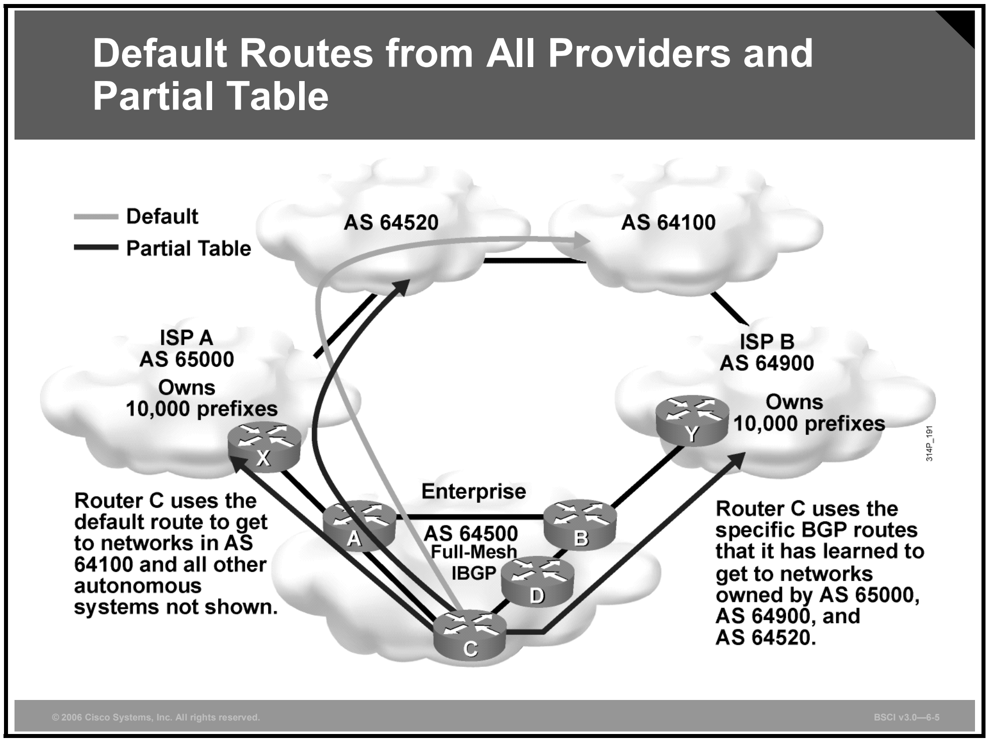
- Path manipulation cannot be performed because only a single route is being received from each ISP.
- Bandwidth manipulation is extremely difficult and can be accomplished only by manipulating the IGP metric of the default route.

- Diverting some of the traffic from one exit point to another is challenging because all destinations are using the same default route for path selection.

Example: Default Routes from All Providers

In the figure, AS 65000 and AS 65250 send default routes into AS 65500. The ISP that a specific router within AS 65500 uses to reach any external address is decided by the IGP metric that is used to reach the default route within the AS.

For example, if you use RIP within AS 65500, router C selects the route with the lowest hop count to the default route when sending packets to network 172.16.0.0.



Default Routes from All Providers and Partial Table

Router C uses the default route to get to networks in AS 64100 and all other autonomous systems not shown.

Router C uses the specific BGP routes that it has learned to get to networks owned by AS 65000, AS 64900, and AS 64520.

In the second design option for multihoming, all ISPs pass default routes and select specific routes to the AS.

An enterprise running EBGP with an ISP that wants a partial routing table generally receives the networks that the ISP and its other customers own. The enterprise can also receive the routes from any other AS.

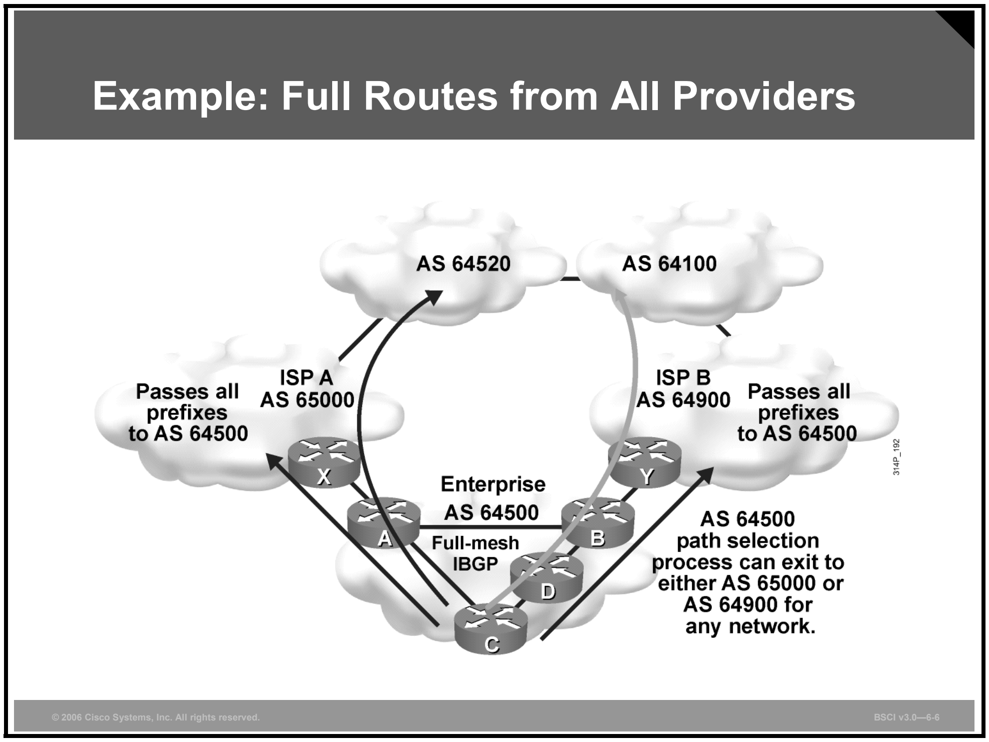
Major ISPs are assigned between 2000 and 10,000 classless interdomain routing (CIDR) blocks of IP addresses from the Internet Assigned Numbers Authority (IANA), which they reassign to their customers. If the ISP passes this information to a customer that wants only a partial BGP routing table, the customer can redistribute these routes into its IGP. The internal routers of the customer (these routers are not running BGP) can then receive these routes via redistribution. They can take the nearest exit point based on the best metric of specific networks instead of taking the nearest exit point based on the default route.

Acquiring a partial BGP table from each provider is beneficial because path selection will be more predictable than when using a default route.

Example: Default Routes from All Providers and Partial Table

In the figure, ISPs in AS 65000 and AS 64900 send default routes and the routes that each ISP owns to AS 64500. The enterprise (AS 64500) asked both providers to also send routes to networks in AS 64520 because of the amount of traffic between AS 64520 and AS 64500.

By running IBGP between the internal routers within AS 64500, AS 64500 can choose the optimal path to reach the customer networks (AS 64520, in this case). The routes to AS 64100 and to other autonomous systems not shown in the figure that are not specifically advertised to AS 64500 by ISP A and ISP B are decided by the IGP metric that is used to reach the default route within the AS.



Example: Full Routes from All Providers

AS 64500 path selection process can exit to either AS 65000 or AS 64900 for any network.

© 2006 Cisco Systems, Inc. All rights reserved.   BSCI v3.0—6-6

In the third multihoming option, all ISPs pass all routes to the AS, and IBGP is run on at least all of the routers in the transit path in this AS. This option allows the internal routers of the AS to take the path through the best ISP for each route.

This configuration requires a lot of resources within the AS because it must process all the external routes.

The AS sends all its routes to the ISPs, which process the routes and pass them to other autonomous systems.

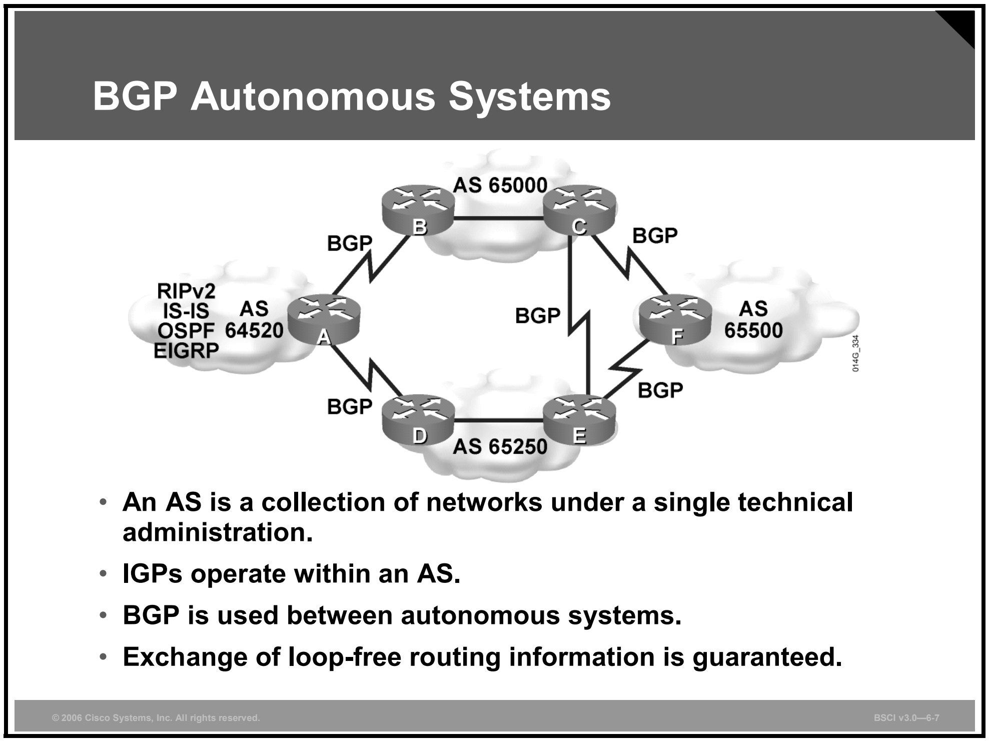
Example: Full Routes from All Providers

In the figure, AS 65000 and AS 64900 send all routes into AS 64500. The ISP that a specific router within AS 64500 uses to reach the external networks is determined by the BGP protocol.

The routers in AS 64500 can be configured to influence the path to certain networks. For example, router A and router B can influence the outbound traffic from AS 64500.

BGP Routing Between Autonomous Systems

This topic describes how BGP routes between autonomous systems.



BGP Autonomous Systems

- An AS is a collection of networks under a single technical administration.
- IGPs operate within an AS.
- BGP is used between autonomous systems.
- Exchange of loop-free routing information is guaranteed.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-7

BGP Is Used Between Autonomous Systems

The main goal of BGP is to provide an interdomain routing system that guarantees loop-free exchange of routing information between autonomous systems. Routers exchange information about paths to destination networks.

BGP is a successor of Exterior Gateway Protocol (EGP), which was developed to isolate networks from each other as the Internet grew.

There are many RFCs relating to BGP4, the current version of BGP, including 1772, 1773, 1774, 1930, 1966, 1997, 1998, 2042, 2385, 2439, 2545, 2547, 2796, 2858, 2918, 3065, 3107, 3392, 4223, and 4271.

BGP4 has many enhancements over earlier protocols. The Internet uses BGP4 extensively to connect ISPs and to connect enterprises to ISPs.

BGP4 and its extensions are the only acceptable versions of BGP available for use on the public-based Internet. BGP4 carries a network mask for each advertised network and supports both variable-length subnet masking (VLSM) and CIDR. BGP4 predecessors did not support these capabilities, which are currently mandatory on the Internet.

When CIDR is being used on a core router for a major ISP, the IP routing table, which is composed mostly of BGP routes, has more than 170,000 CIDR blocks. Not using CIDR at the Internet level would cause the IP routing table to have more than 2,000,000 entries. Using BGP4 and CIDR prevents the Internet routing table from becoming too large for interconnecting millions of users.

AS Numbers

Recall that an AS is a collection of networks under a single technical administration. IGPs operate within an AS, and BGP (specifically BGP4) is used between autonomous systems on the Internet.

The IANA is the organization responsible for allocating AS numbers. Specifically, the American Registry for Internet Numbers (ARIN) has the jurisdiction to assign numbers for the Americas, the Caribbean, and Africa. Réseaux IP Européens Network Coordination Center (RIPE NCC) administers AS numbers for Europe, and the Asia Pacific Network Information Center (APNIC) administers the numbers for the Asia-Pacific region.

AS numbers are 16-bit numbers ranging from 1 to 65535; RFC 1930 provides guidelines for the use of AS numbers. A range of AS numbers, 64512 through 65535, is reserved for private use, much like private IP addresses. The AS numbers used in this course are all in the private range to avoid publishing AS numbers belonging to organizations.

Note Using an IANA-assigned AS number rather than a private AS number is necessary only if your organization plans to use an EGP, such as BGP, and connect to a public network, such as the Internet.

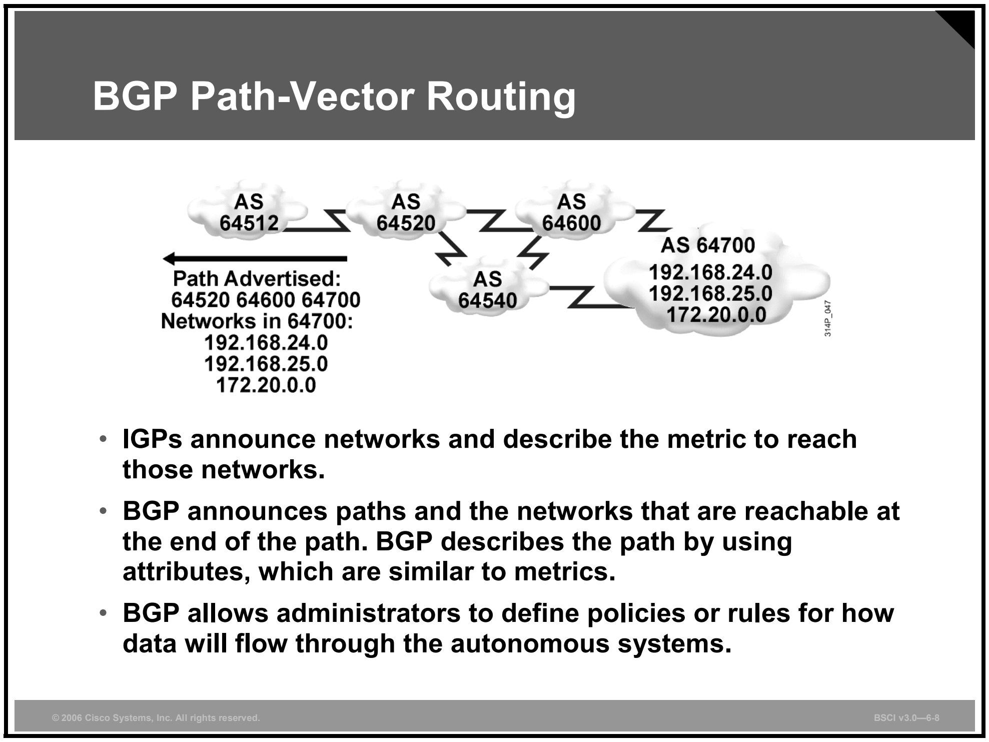
Comparison with IGPs

BGP works differently than IGPs. An internal routing protocol looks for the quickest path from one point in a corporate network to another based on certain metrics. RIP uses hop counts that look to cross the fewest Layer 3 devices to reach the destination network. OSPF and EIGRP look for the best speed according to the bandwidth statement on the interface. All internal routing protocols look at the path cost to a destination.

In contrast, BGP, an external routing protocol, does not look at speed for the best path. Rather, BGP is a policy-based routing (PBR) protocol that allows an AS to control traffic flow using multiple BGP path attributes. BGP allows a provider to fully use all its bandwidth by manipulating these path attributes.

Path-Vector Functionality

This topic describes how BGP uses path-vector functionality.



BGP Path-Vector Routing

- **IGPs announce networks and describe the metric to reach those networks.**
- **BGP announces paths and the networks that are reachable at the end of the path. BGP describes the path by using attributes, which are similar to metrics.**
- **BGP allows administrators to define policies or rules for how data will flow through the autonomous systems.**

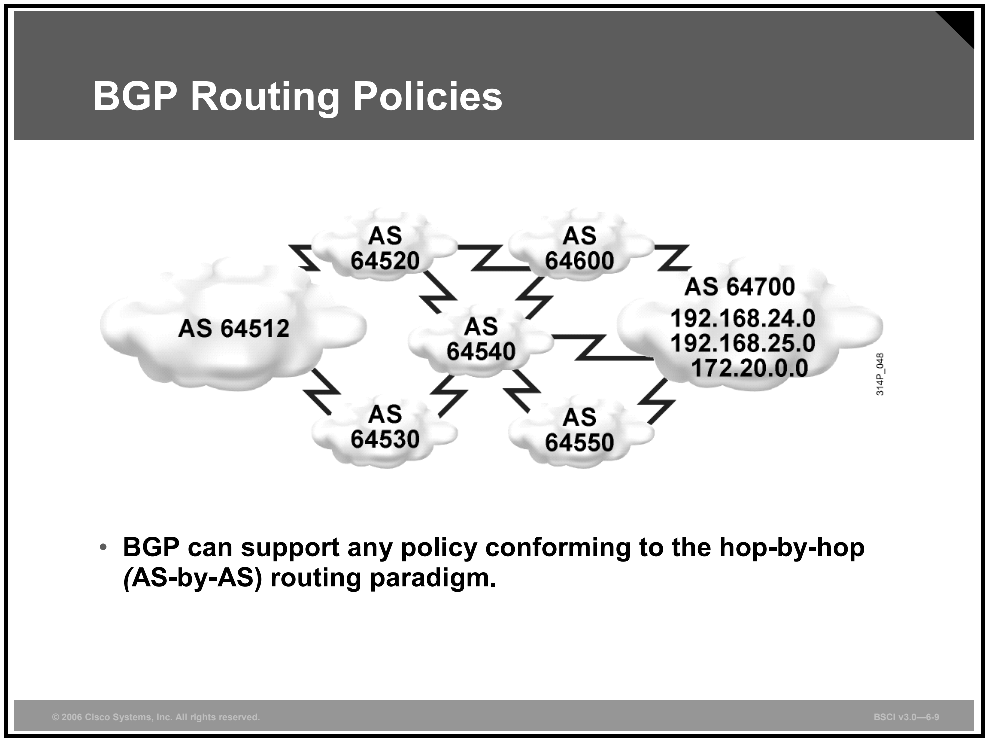
BSCI v3.0—6-8

Internal routing protocols announce a list of networks and the metrics to get to each network. In contrast, BGP routers exchange network reachability information, called path vectors, made up of path attributes. The path-vector information includes a list of the full path of BGP AS numbers (hop by hop) necessary to reach a destination network and the networks that are reachable at the end of the path.

Other attributes include the IP address to get to the next AS (the next-hop attribute) and an indication of how the networks at the end of the path were introduced into BGP (the origin code attribute).

This AS path information is useful to construct a graph of loop-free autonomous systems and is used to identify routing policies so that restrictions on routing behavior can be enforced based on the AS path.

The AS path is always loop-free. A router running BGP does not accept a routing update that already includes the router AS number in the path list, because the update has already passed through its AS, and accepting it again would result in a routing loop.



BGP Routing Policies

- **BGP can support any policy conforming to the hop-by-hop (AS-by-AS) routing paradigm.**

BGP allows routing-policy decisions at the AS level to be enforced. These policies can be implemented for all networks owned by an AS, for a certain CIDR block of network numbers (prefixes), or for individual networks or subnetworks.

BGP specifies that a BGP router can advertise to neighboring autonomous systems only those routes that it uses itself. This rule reflects the hop-by-hop routing paradigm that the Internet generally uses.

The hop-by-hop routing paradigm does not support all possible policies. For example, BGP does not enable one AS to send traffic to a neighboring AS intending that the traffic take a different route from that taken by traffic that originates in that neighboring AS. In other words, you cannot influence how a neighboring AS routes traffic, but you can influence how your traffic gets to a neighboring AS. However, BGP supports any policy that conforms to the hop-by-hop routing paradigm.

Because the Internet currently uses the hop-by-hop routing paradigm only, and because BGP can support any policy that conforms to that paradigm, BGP is highly applicable as an inter-AS routing protocol.

Example: BGP Routing Policies

For example, in the figure, the following paths are possible for AS 64512 to reach networks in AS 64700 through AS 64520:

- 64520 64600 64700
- 64520 64600 64540 64550 64700
- 64520 64540 64600 64700
- 64520 64540 64550 64700

AS 64512 does not see all these possibilities.

AS 64520 advertises to AS 64512 only its best path, 64520 64600 64700, in the same way that IGPs announce only their best least-cost routes. This path is the only path through AS 64520 that AS 64512 sees. All packets that are destined for 64700 through 64520 will take this path.

Even though other paths exist, AS 64512 can only use what AS 64520 advertises for the networks in AS 64700. The AS path that is advertised, 64520 64600 64700, is the AS-by-AS (hop-by-hop) path that AS 64520 will use to reach the networks in AS 64700. AS 64520 will not announce another path, such as 64520 64540 64600 64700, because it did not choose that as the best path based on the BGP routing policy in AS 64520.

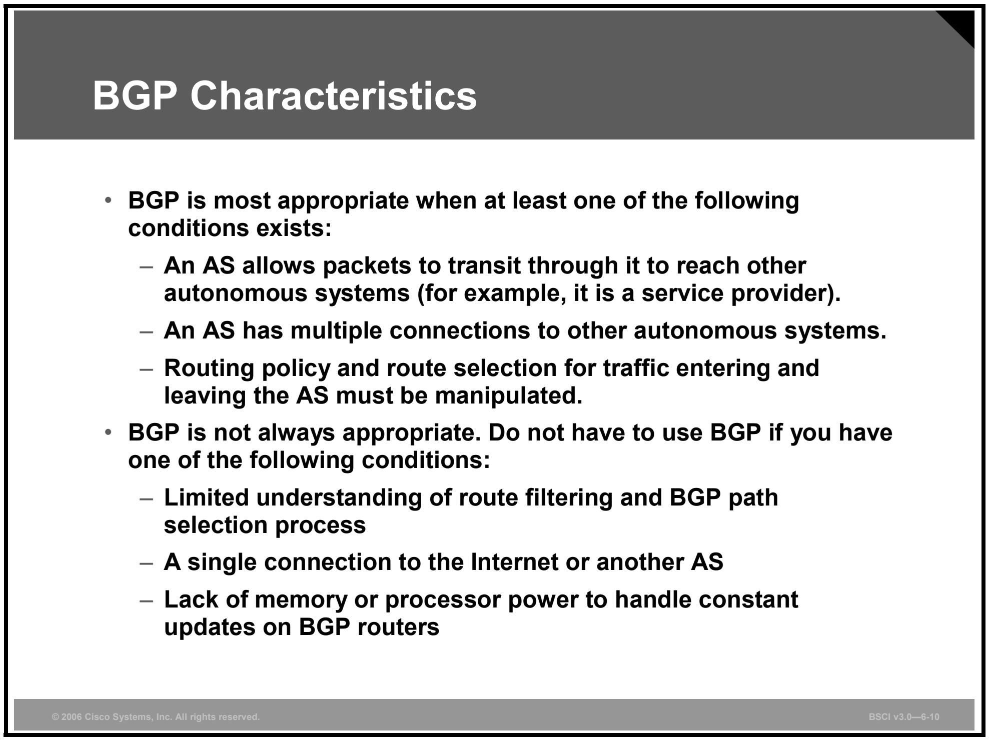
AS 64512 will not learn of the second-best path or any other paths from AS 64520 unless the best path of AS 64520 becomes unavailable.

Even if AS 64512 were aware of another path through AS 64520 and wanted to use it, AS 64520 would not route packets along that other path because AS 64520 selected 64520 64600 64700 as its best path and all AS 64520 routers will use that path as a matter of BGP policy. BGP does not let one AS send traffic to a neighboring AS, intending that the traffic take a different route from that taken by traffic originating in the neighboring AS.

To reach the networks in AS 64700, AS 64512 can choose to use AS 64520 or it can choose to go through the path that AS 64530 is advertising. AS 64512 selects the best path to take based on its own BGP routing policies.

Features of BGP

This topic describes the features of BGP in terms of deployment, enhancements over other distance vector routing protocols, and database types.



BGP Characteristics

- BGP is most appropriate when at least one of the following conditions exists:
  - An AS allows packets to transit through it to reach other autonomous systems (for example, it is a service provider).
  - An AS has multiple connections to other autonomous systems.
  - Routing policy and route selection for traffic entering and leaving the AS must be manipulated.
- BGP is not always appropriate. Do not have to use BGP if you have one of the following conditions:
  - Limited understanding of route filtering and BGP path selection process
  - A single connection to the Internet or another AS
  - Lack of memory or processor power to handle constant updates on BGP routers

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-10

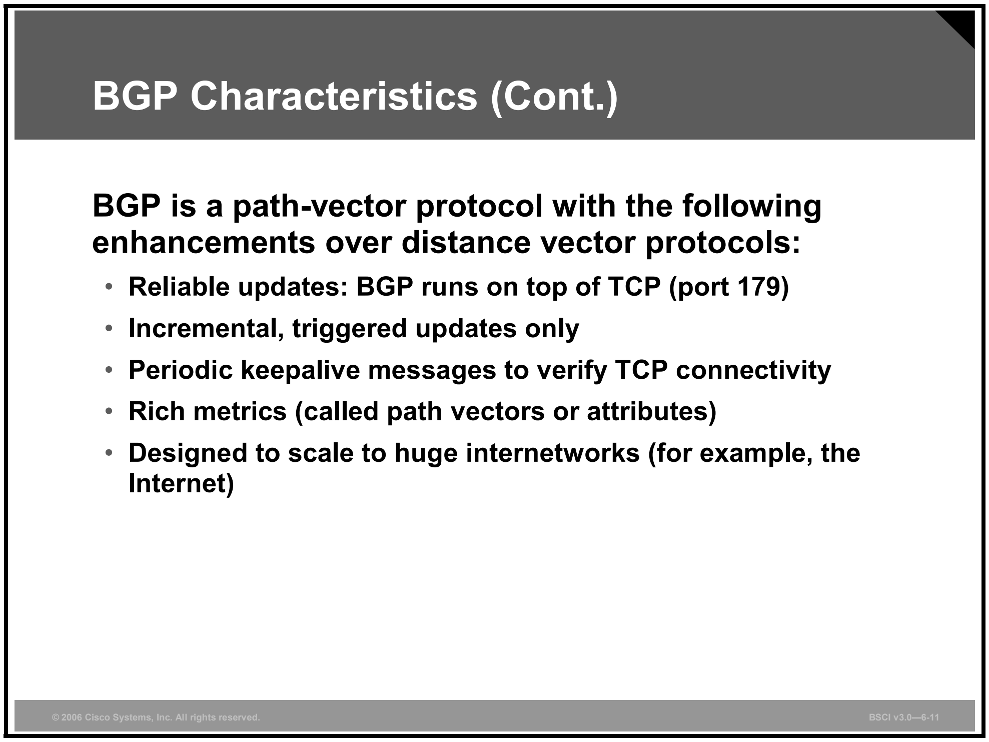
BGP allows ISPs to communicate and exchange packets. The ISPs have multiple connections to each other and agreements to exchange updates. BGP is used to implement the agreements between two or more autonomous systems.

Improper controlling and filtering of BGP updates can potentially allow an outside AS to affect the traffic flow to your AS. It is important to know how BGP operates and how to configure it properly to prevent this situation.

For example, if you are a customer connected to ISP-A and ISP-B (for redundancy), you want to implement a routing policy to ensure that ISP-A does not send traffic to ISP-B via your AS. You do not want to waste valuable resources and bandwidth within your AS to route traffic for your ISPs, but you want to be able to receive traffic destined to your AS through each ISP.

BGP is not always an appropriate solution to interconnect autonomous systems. For example, if only one exit path from the AS exists, a default route is the most appropriate solution. In this case, BGP would unnecessarily use router CPU resources and memory.

If the routing policy that you implement in an AS is consistent with the policy in the ISP AS, it is not necessary or desirable to configure BGP in that AS.



BGP Characteristics (Cont.)

BGP is a path-vector protocol with the following enhancements over distance vector protocols:

- **Reliable updates: BGP runs on top of TCP (port 179)**
- **Incremental, triggered updates only**
- **Periodic keepalive messages to verify TCP connectivity**
- **Rich metrics (called path vectors or attributes)**
- **Designed to scale to huge internetworks (for example, the Internet)**

BGP is categorized as an advanced distance vector protocol, but it is actually a path-vector protocol. BGP is very different from standard distance vector protocols like RIP.

BGP uses TCP as its transport protocol, which provides connection-oriented reliable delivery. BGP assumes that its communication is reliable; therefore, it does not have to implement retransmission or error recovery mechanisms. BGP uses TCP port 179. Two routers using BGP form a TCP connection with one another and exchange messages to open and confirm the connection parameters. These two BGP routers are called peer routers, or neighbors.

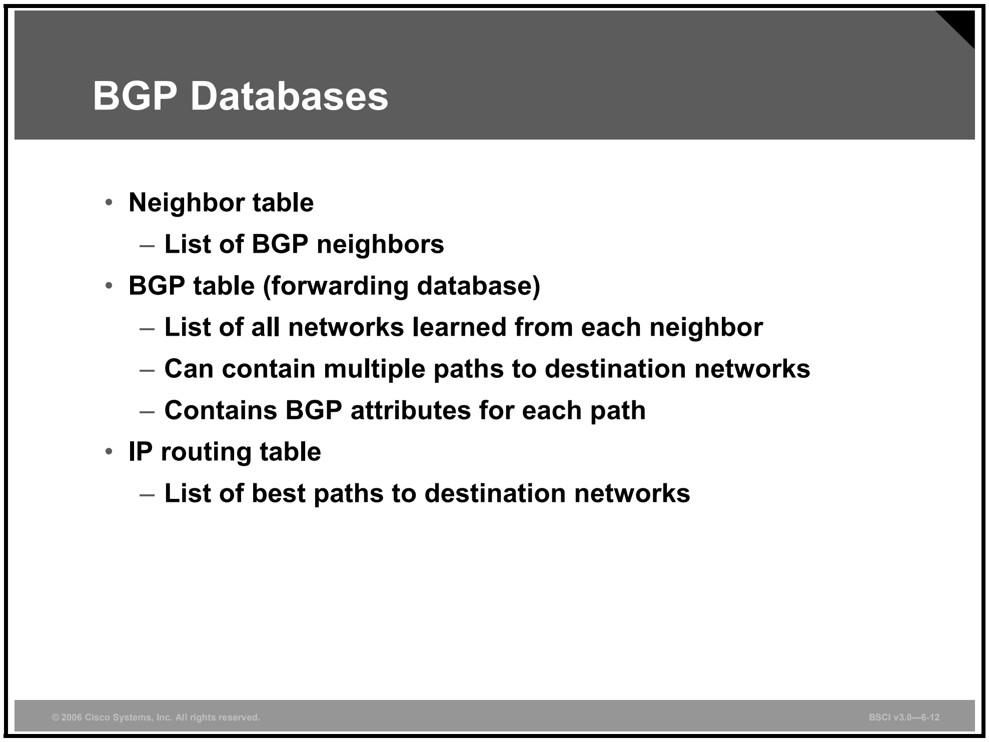
After the connection is made, BGP peers exchange full routing tables. However, because the connection is reliable, BGP peers subsequently send only changes (incremental, or triggered, updates) after that. Reliable links do not require periodic routing updates; therefore, routers use triggered updates instead. BGP sends keepalive messages, similar to the hello messages sent by OSPF, IS-IS, and EIGRP.

BGP is the only IP routing protocol to use TCP as its transport layer. OSPF, IGRP, and EIGRP reside directly above the IP layer, and RIP version 1 (RIPv1) and RIP version 2 (RIPv2) use User Datagram Protocol (UDP) for their transport layer.

OSPF and EIGRP have their own internal function to ensure that update packets are explicitly acknowledged. These protocols use a one-for-one window so that if either OSPF or EIGRP has multiple packets to send, the next packet cannot be sent until they receive an acknowledgment from the first update packet. This process can be very inefficient and cause latency issues if thousands of update packets must be exchanged over relatively slow serial links. OSPF and EIGRP rarely have thousands of update packets to send. EIGRP can hold more than 100 networks in one EIGRP update packet, so 100 EIGRP update packets can hold up to 10,000 networks, and most organizations do not have 10,000 subnets in the enterprise.

BGP, on the other hand, has more than 170,000 networks (and growing) on the Internet to advertise and it uses TCP to handle the acknowledgment function. TCP uses a dynamic window, which allows 65,576 bytes to be outstanding before it stops and waits for an acknowledgment. For example, if 1000-byte packets are being sent, BGP would stop and wait for an acknowledgment only when 65 packets had not been acknowledged, when using the maximum window size.

TCP is designed to use a sliding window, where the receiver will acknowledge at the halfway point of the sending window. This method allows any TCP application, such as BGP, to continue to stream packets without having to stop and wait, as OSPF or EIGRP would require.



BGP Databases

- **Neighbor table**
  - **List of BGP neighbors**
- **BGP table (forwarding database)**
  - **List of all networks learned from each neighbor**
  - **Can contain multiple paths to destination networks**
  - **Contains BGP attributes for each path**
- **IP routing table**
  - **List of best paths to destination networks**

BSCI v3.0—6-12

A router running BGP keeps its own tables to store BGP information that it receives from and sends to other routers, including a neighbor table, a BGP table (also called a forwarding database or topology database), and an IP routing table.

For BGP to establish an adjacency, you must configure it explicitly for each neighbor. BGP forms a TCP relationship with each of the configured neighbors and keeps track of the state of these relationships by periodically sending a BGP/TCP keepalive message.

Note The BGP sends BGP/TCP keepalives by default every 60 seconds.

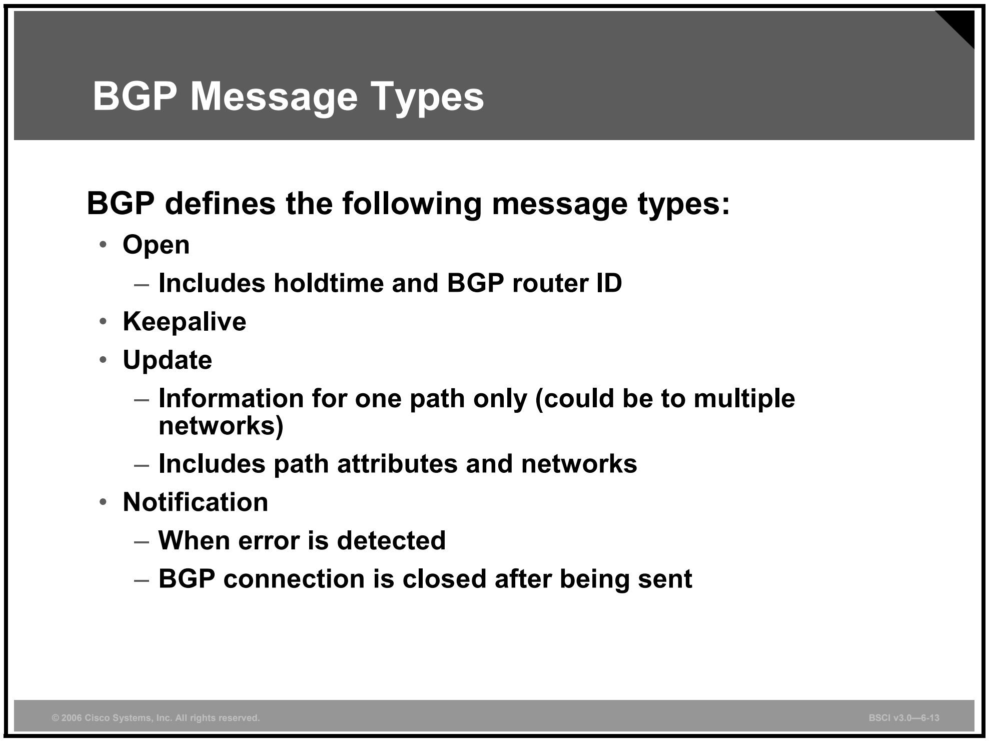
After establishing an adjacency, the neighbors exchange the BGP routes that are in their IP routing table. Each router collects these routes from each neighbor that successfully establishes an adjacency and then places them in its BGP forwarding database. All routes that have been learned from each neighbor are placed into the BGP forwarding database. The best routes for each network are selected from the BGP forwarding database using the BGP route selection process and then offered to the IP routing table.

Each router compares the offered BGP routes to any other possible paths to those networks, and the best route, based on administrative distance, is installed in the IP routing table.

EBGP routes (BGP routes learned from an external AS) have an administrative distance of 20. IBGP routes (BGP routes learned from within the AS) have an administrative distance of 200.

BGP Message Types

This topic describes the functionality of each BGP message type.



BGP Message Types

BGP defines the following message types:

- **Open**
  - **Includes holdtime and BGP router ID**
- **Keepalive**
- **Update**
  - **Information for one path only (could be to multiple networks)**
  - **Includes path attributes and networks**
- **Notification**
  - **When error is detected**
  - **BGP connection is closed after being sent**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-13

The four BGP message types are open, keepalive, update, and notification.

After a TCP connection is established, the first message sent by each side is an open message. If the open message is acceptable, the side that receives the message sends a keepalive message confirming the open message. After the receiving side confirms the open message and establishes the BGP connection, the BGP peers can exchange any update, keepalive, and notification messages.

BGP peers initially exchange their full BGP routing tables. Incremental updates are sent only after topology changes in the network. BGP peers send keepalive messages to ensure that the connection between the BGP peers still exists; they send notification packets in response to errors or special conditions.

Here are more details about the different types of BGP messages:

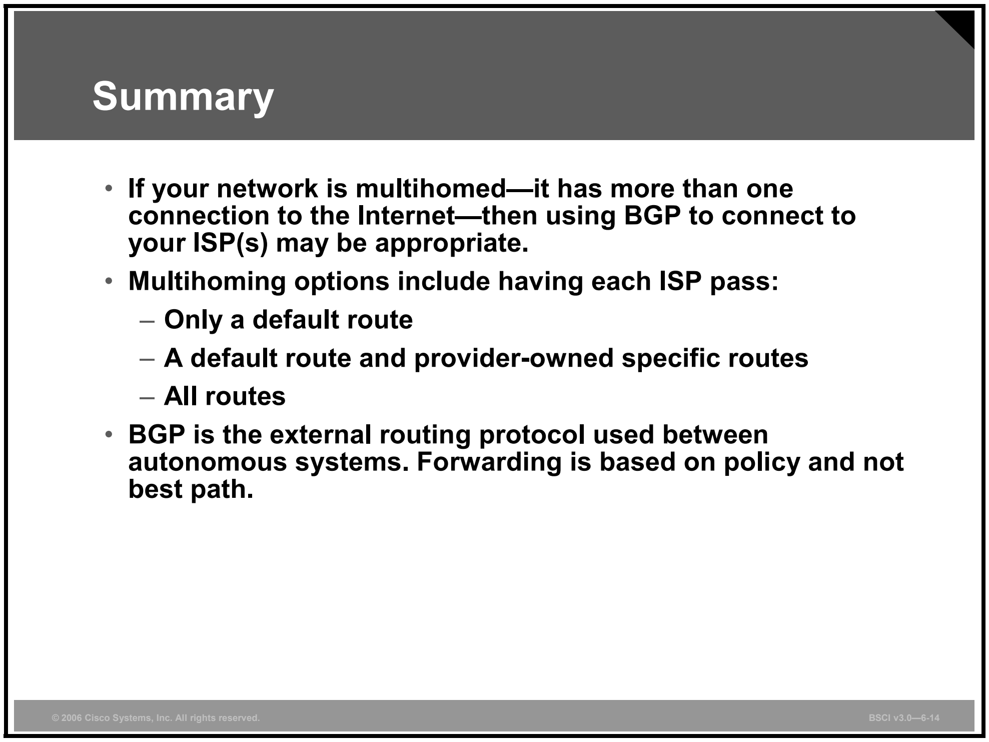
- **Open message:** An open message includes the following information:
  - **Version number:** The suggested version number. The highest common version that both routers support is used. Most BGP implementations today use BGP4.
  - **AS number:** The AS number of the local router. The peer router verifies this information. If it is not the AS number that is expected, the BGP session is torn down.
  - **Hold time:** Maximum number of seconds that can elapse between the successive keepalive and update messages from the sender. On receipt of an open message, the router calculates the value of the hold timer by using whichever is smaller: its configured hold time or the hold time that was received in the open message.

- **BGP router ID:** This 32-bit field indicates the BGP ID of the sender. The BGP ID is an IP address that is assigned to that router, and it is determined at startup. The BGP router ID is chosen in the same way that the OSPF router ID is chosen: it is the highest active IP address on the router unless a loopback interface with an IP address exists. In this case, the router ID is the highest loopback IP address. The router ID can also be statically configured.
- **Optional parameters:** These parameters are Type, Length, and Value (TLV)-encoded. An example of an optional parameter is session authentication.

- **Keepalive message:** BGP keepalive messages are exchanged between BGP peers often enough to keep the hold timer from expiring. If the negotiated hold-time interval is 0, then periodic keepalive messages are not sent. A keepalive message consists of only a message header.
- **Update message:** A BGP update message has information on one path only; multiple paths require multiple update messages. All the attributes in the update message refer to that path, and the networks are those that can be reached through it. An update message can include the following fields:
  - **Withdrawn routes:** This list displays IP address prefixes for routes that are withdrawn from service, if any.
  - **Path attributes:** These attributes include the AS path, origin, local preference, and so on (as described later in this module). Each path attribute includes the attribute TLV. The attribute type consists of the attribute flags, followed by the attribute type code.
  - **Network-layer reachability information:** This field contains a list of IP address prefixes that are reachable by this path.
- **Notification message:** A BGP notification message is sent when an error condition is detected; the BGP connection is closed immediately after this is sent. Notification messages include an error code, an error subcode, and data that are related to the error.

Summary

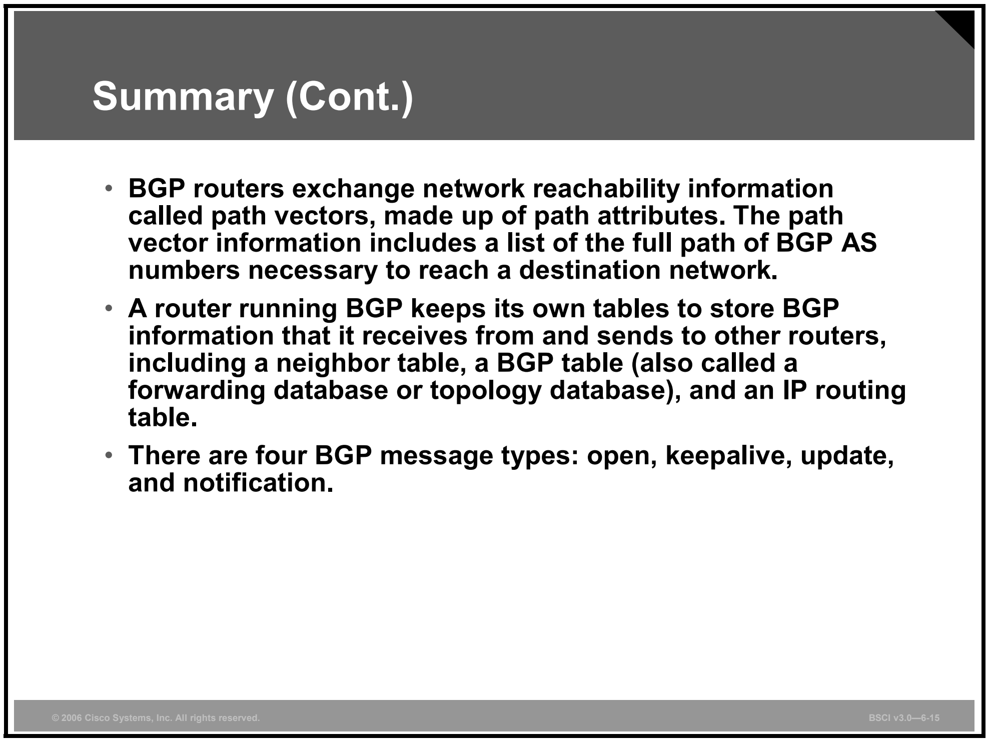
This topic summarizes the key points that were discussed in this lesson.



Summary

- If your network is multihomed—it has more than one connection to the Internet—then using BGP to connect to your ISP(s) may be appropriate.
- Multihoming options include having each ISP pass:
  - Only a default route
  - A default route and provider-owned specific routes
  - All routes
- BGP is the external routing protocol used between autonomous systems. Forwarding is based on policy and not best path.

BSCI v3.0—6-14



Summary (Cont.)

- BGP routers exchange network reachability information called path vectors, made up of path attributes. The path vector information includes a list of the full path of BGP AS numbers necessary to reach a destination network.
- A router running BGP keeps its own tables to store BGP information that it receives from and sends to other routers, including a neighbor table, a BGP table (also called a forwarding database or topology database), and an IP routing table.
- There are four BGP message types: open, keepalive, update, and notification.

BSCI v3.0—6-15

Lesson 2

Explaining EBGP and IBGP

Overview

This lesson discusses important terminology that is used in establishing Border Gateway Protocol (BGP) peering relationships. The following terms are explained: BGP speaker, BGP router, BGP neighbor, and BGP peer.

This lesson defines External BGP (EBGP) and Internal BGP (IBGP) neighbors and the requirements for establishing these relationships. In addition, this lesson examines the difference between an interior gateway protocol (IGP) and BGP and explains the reason for having all routers in the transit path within an AS running IBGP.

Understanding the relationship between various types of BGP routers and the common terminology that is used when you are discussing these routers is necessary for troubleshooting connectivity issues between BGP neighbors.

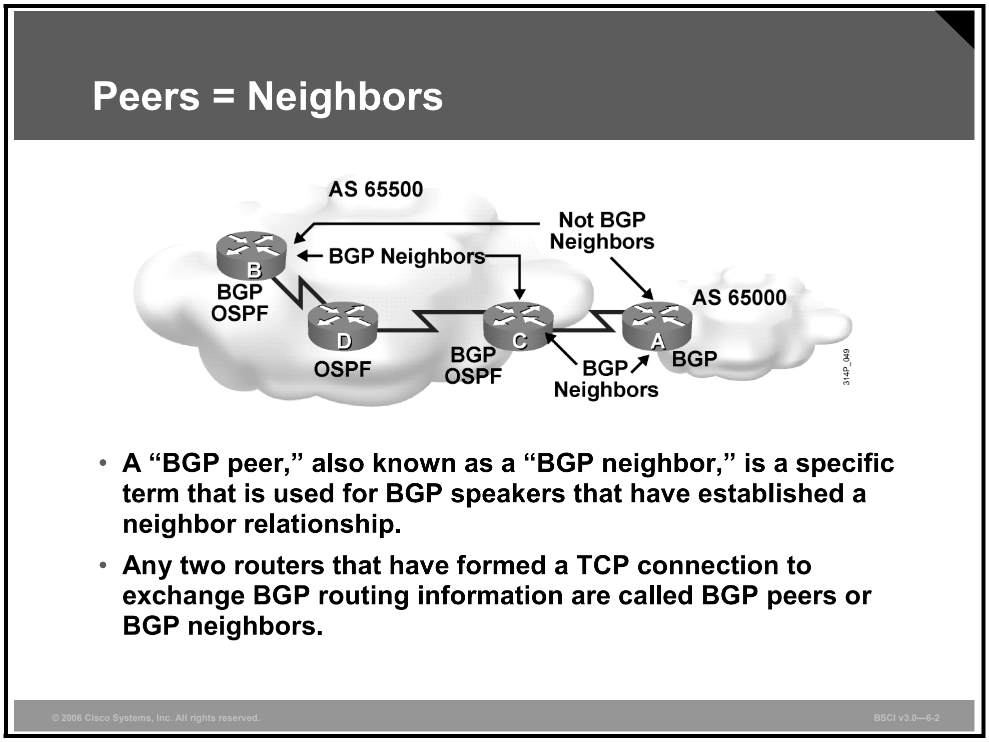
Objectives

Upon completing this lesson, you will be able to describe the structural concepts of a BGP network. This ability includes being able to meet these objectives:

- Define terms used to describe BGP routers and their relationships
- Describe the requirements for establishing an EBGP neighbor relationship
- Describe the requirements for establishing an IBGP neighbor relationship
- Explain why IBGP route propagation requires all routers in the transit path in an AS to run IBGP

BGP Neighbor Relationships

This topic defines terms used to describe BGP routers and their relationships.



Peers = Neighbors

- **A “BGP peer,” also known as a “BGP neighbor,” is a specific term that is used for BGP speakers that have established a neighbor relationship.**
- **Any two routers that have formed a TCP connection to exchange BGP routing information are called BGP peers or BGP neighbors.**

BSCI v3.0—6-2

No one router can handle communications with all the routers that run BGP. There are tens of thousands of routers that run BGP and are connected to the Internet, representing more than 21,000 autonomous systems.

A BGP router forms a direct neighbor relationship with a limited number of other BGP routers. Through these BGP neighbors, a BGP router learns of the paths through the Internet to reach any advertised network. Any router that runs BGP is known as a BGP speaker.

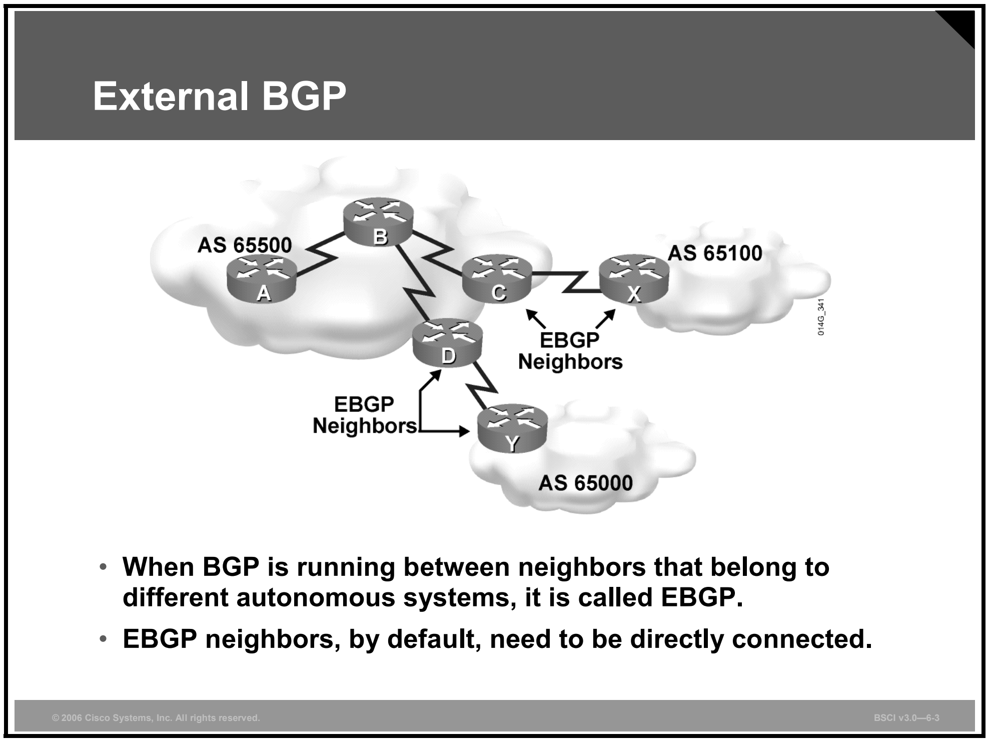
The term “BGP peer” has a specific meaning: a BGP speaker that is configured to form a neighbor relationship with another BGP speaker for the purpose of directly exchanging BGP routing information with each other. A BGP speaker has a limited number of BGP neighbors with which it peers and forms a TCP-based relationship.

BGP peers are also known as BGP neighbors and can be either internal or external to the AS.

A BGP peer must be configured with a BGP **neighbor** command. The administrator instructs the BGP speaker to establish a relationship with the address listed in the **neighbor** command and to exchange the BGP routing updates with that neighbor.

Establishing EBGP Neighbor Relationships

This topic describes the requirements for establishing an external BGP neighbor relationship.



External BGP

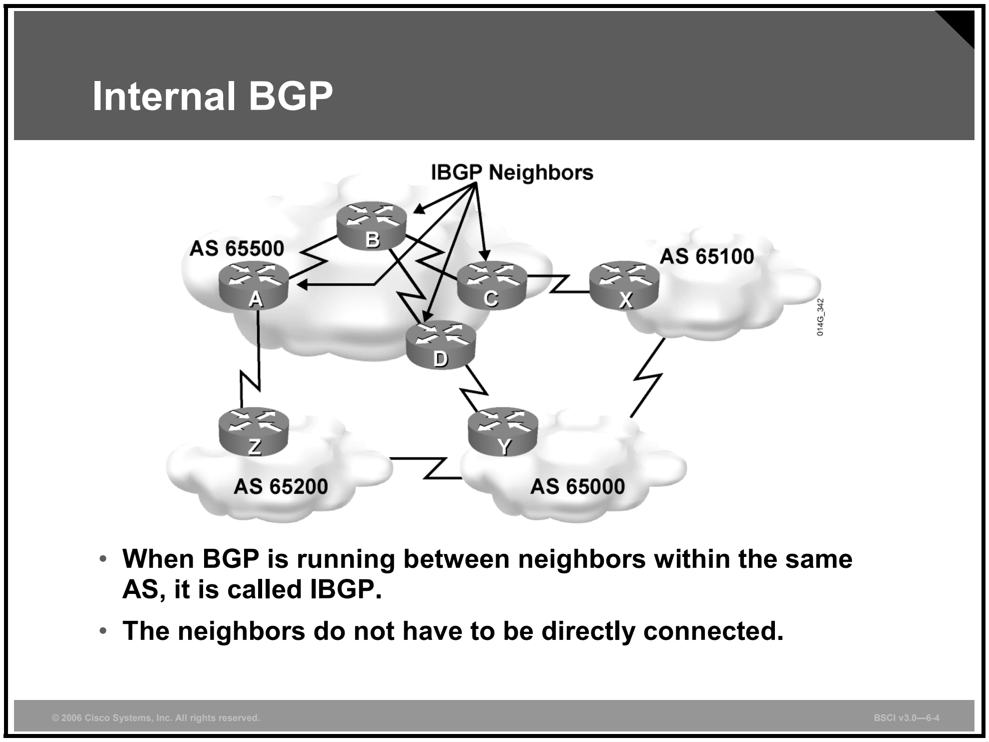
- When BGP is running between neighbors that belong to different autonomous systems, it is called EBGP.
- EBGP neighbors, by default, need to be directly connected.

Recall that when BGP is running between routers in different autonomous systems, it is called EBGP. By default, routers running EBGP are directly connected to each other.

An EBGP neighbor is a router outside this AS; an IGP is not run between the EBGP neighbors. For two routers to exchange BGP routing updates, the TCP-reliable transport layer on each side must successfully pass the TCP three-way handshake before the BGP session can be established. Therefore, the IP address used in the BGP **neighbor** command must be reachable without using an IGP, which can be accomplished by pointing at an address that is reachable through a directly connected network or by using static routes to that IP address. Generally, the neighbor address that is used is the address on a directly connected network.

Establishing IBGP Neighbor Relationships

This topic describes the requirements for establishing an internal BGP neighbor relationship.



Internal BGP

- When BGP is running between neighbors within the same AS, it is called IBGP.
- The neighbors do not have to be directly connected.

BSCI v3.0—6-4

Recall that BGP that runs between routers within the same AS is called IBGP. IBGP runs within an AS to exchange BGP information so that all BGP speakers have the same BGP routing information about outside autonomous systems.

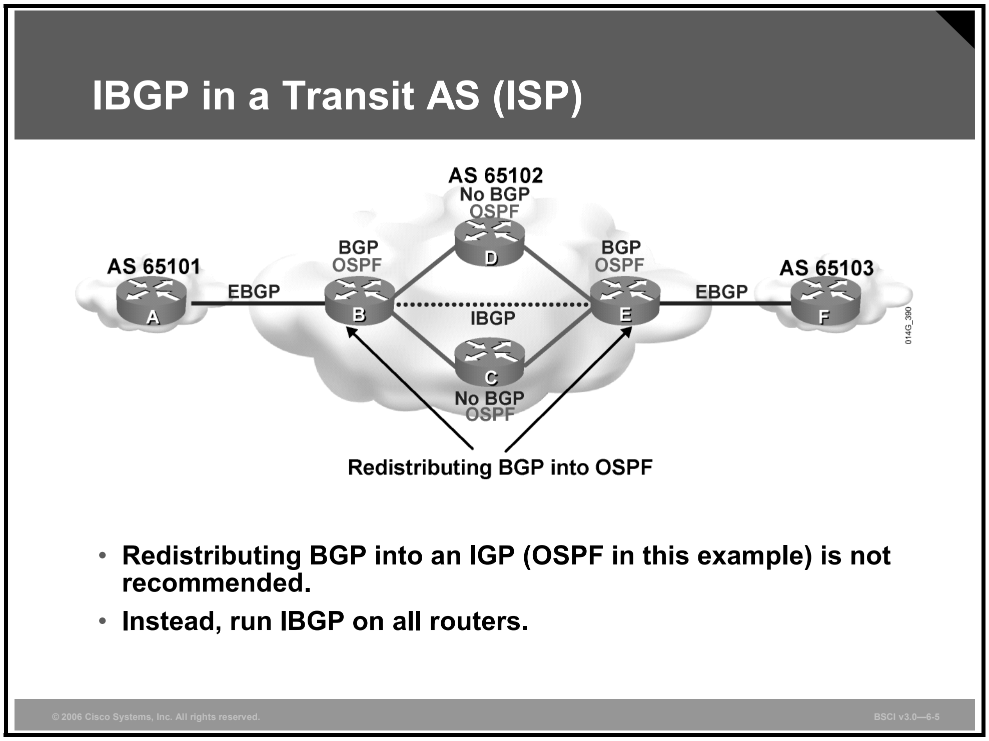
Routers running IBGP do not have to be directly connected to each other as long as they can reach each other so that TCP handshaking can be performed to set up the BGP neighbor relationships. The IBGP neighbor can be reached by a directly connected network, static routes, or by the internal routing protocol. Because multiple paths generally exist within an AS to reach the other IBGP routers, a loopback address is generally used in the BGP **neighbor** command to establish the IBGP sessions.

Example: Internal BGP

When multiple routers in an AS are running BGP, they exchange BGP routing updates with one another. In the figure, routers A, D, and C learn the paths to the external autonomous systems from their respective EBGP neighbors (Z, Y, and X). If the link between D and Y goes down, D must learn new routes to the external autonomous systems. Other BGP routers within AS 65500 that were using D to get to external networks must also be informed that the path through D is not available. Those BGP routers within AS 65500 need to have the alternate paths through routers A and C in their BGP forwarding database. You must set up IBGP sessions between all BGP routers in AS 65500, so each router within the AS learns about paths to the external networks via IBGP.

IBGP on All Routers in Transit Path

This topic explains why IBGP route propagation requires all routers in the transit path in an AS to run IBGP.



IBGP in a Transit AS (ISP)

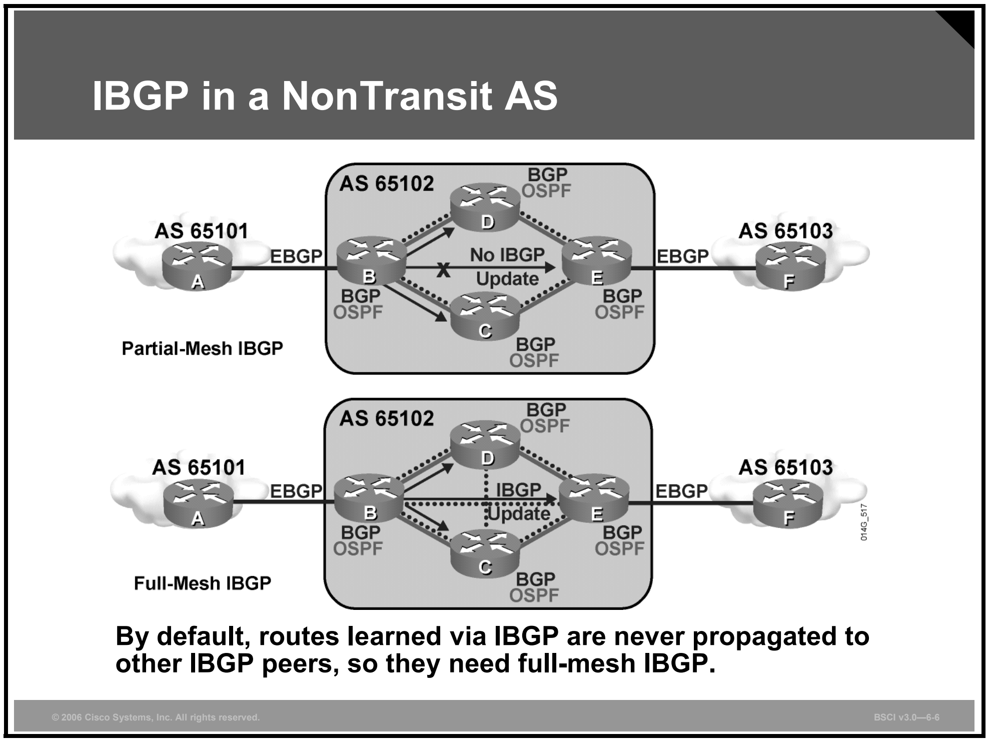
- Redistributing BGP into an IGP (OSPF in this example) is not recommended.
- Instead, run IBGP on all routers.

BSCI v3.0—6-5

IBGP in a Transit AS

BGP was originally intended to run along the borders of an AS with the routers in the middle of the AS ignorant of the details of BGP (hence the name "*Border* Gateway Protocol"). A transit AS, such as the one in the figure, is an AS that routes traffic from one external AS to another external AS. Typically, transit autonomous systems are Internet service providers (ISPs). All routers in a transit AS must have complete knowledge of external routes. Theoretically, one way to achieve this goal is to redistribute BGP routes into an IGP at the edge routers. However, this approach has problems.

Because the current Internet routing table is very large, redistributing all the BGP routes into an IGP is not a scalable method for the interior routers within an AS to learn about the external networks. Another method that you can use is to run IBGP within the AS.



IBGP in a NonTransit AS

By default, routes learned via IBGP are never propagated to other IBGP peers, so they need full-mesh IBGP.

BSCI v3.0—6-6

IBGP in a Nontransit AS

A nontransit AS, such as an organization that is multihoming with two ISPs, does not pass routes between the ISPs. However, the BGP routers within the AS still require knowledge of all BGP routes passed to the AS to make proper routing decisions.

BGP does not work in the same manner as IGPs. Because the designers of BGP could not guarantee that an AS would run BGP on all routers, a method had to be developed to ensure that IBGP speakers could pass updates to one another while ensuring that no routing loops would exist.

To avoid routing loops within an AS, BGP specifies that routes learned through IBGP are never propagated to other IBGP peers.

Recall that the **neighbor** command enables BGP updates between BGP speakers. By default, each BGP speaker is assumed to have a neighbor statement for all other IBGP speakers in the AS, which is known as full mesh-IBGP.

If the sending IBGP neighbor is not fully meshed with each IBGP router, the routers that are not peering with this router will have different IP routing tables from the routers that are peering with it. The inconsistent routing tables can cause routing loops or routing black holes, because the default assumption by all routers running BGP within an AS is that each BGP router is exchanging IBGP information directly with all other BGP routers in the AS.

If all IBGP neighbors are fully meshed, when a change is received from an external AS, the BGP router for the local AS is responsible for informing all other IBGP neighbors of the change. IBGP neighbors that receive this update do not send it to any other IBGP neighbor, because they assume that the sending IBGP neighbor is fully meshed with all other IBGP speakers and has sent each IBGP neighbor the update.

Example: IBGP Partial Mesh

The top portion of the figure shows IBGP update behavior in a partially meshed neighbor environment.

Router B receives a BGP update from router A. Router B has two IBGP neighbors, routers C and D, but does not have an IBGP neighbor relationship with router E. Routers C and D learn about any networks that were added or withdrawn behind router B. Even if routers C and D have IBGP neighbor sessions with router E, they assume that the AS is fully meshed for IBGP and do not replicate the update and send it to router E.

Sending an IBGP update to router E is the responsibility of router B because it is the router with first-hand knowledge of the networks in and beyond AS 65101. Router E does not learn of any networks through router B and will not use router B to reach any networks in AS 65101 or other autonomous systems behind AS 65101.

Example: IBGP Full Mesh

In the lower portion of the figure, IBGP is fully meshed. When router B receives an update from router A, it updates all three of its IBGP peers, router C, router D, and router E. The IGP is used to route the TCP segment containing the BGP update from router A to router E as the routers are not directly connected. The update is sent once to each neighbor and not duplicated by any other IBGP neighbor, which reduces unnecessary traffic. In fully meshed IBGP, each router assumes that every other internal router has a neighbor statement that points to each IBGP neighbor.

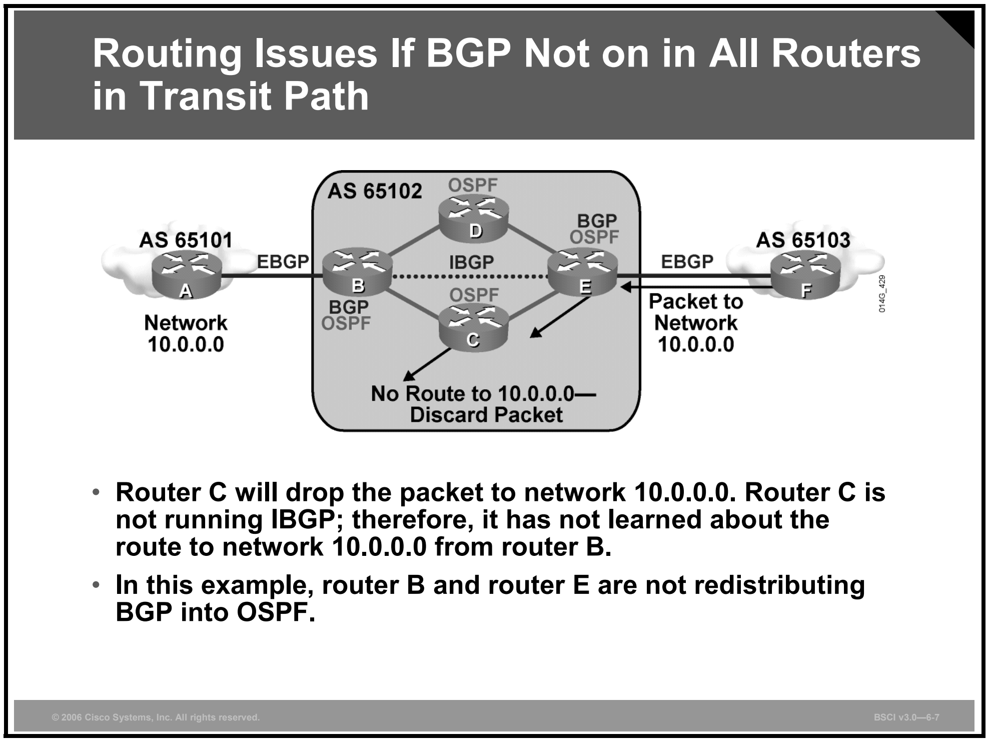
TCP and Full Mesh

TCP was selected as the transport layer for BGP because TCP can move a large volume of data reliably. With the very large full Internet routing table changing constantly, using TCP for windowing and reliability was determined to be the best solution, as opposed to developing a BGP one-for-one windowing capability like OSPF or EIGRP.

TCP sessions cannot be multicast or broadcast because TCP has to ensure the delivery of packets to each recipient. Because TCP cannot use broadcasting, BGP cannot use it either.

Since each IBGP router needs to send routes to all the other IBGP neighbors in the same AS (so that they all have a complete picture of the routes sent to the AS) and they cannot use broadcast, they must use fully meshed BGP (TCP) sessions.

When all routers running BGP in an AS are fully meshed and have the same database as a result of a consistent routing policy, they can apply the same path selection formula. The path selection results will therefore be uniform across the AS. Uniform path selection across the AS means no routing loops and a consistent policy for exiting and entering the AS.



Routing Issues If BGP Not on in All Routers in Transit Path

- Router C will drop the packet to network 10.0.0.0. Router C is not running IBGP; therefore, it has not learned about the route to network 10.0.0.0 from router B.
- In this example, router B and router E are not redistributing BGP into OSPF.

BSCI v3.0—6-7

Example: Routing Issues if BGP Is Not on in All Routers in Transit Path

All routers in the path between IBGP neighbors, known as the transit path, must also be running BGP, as illustrated by the example in the figure.

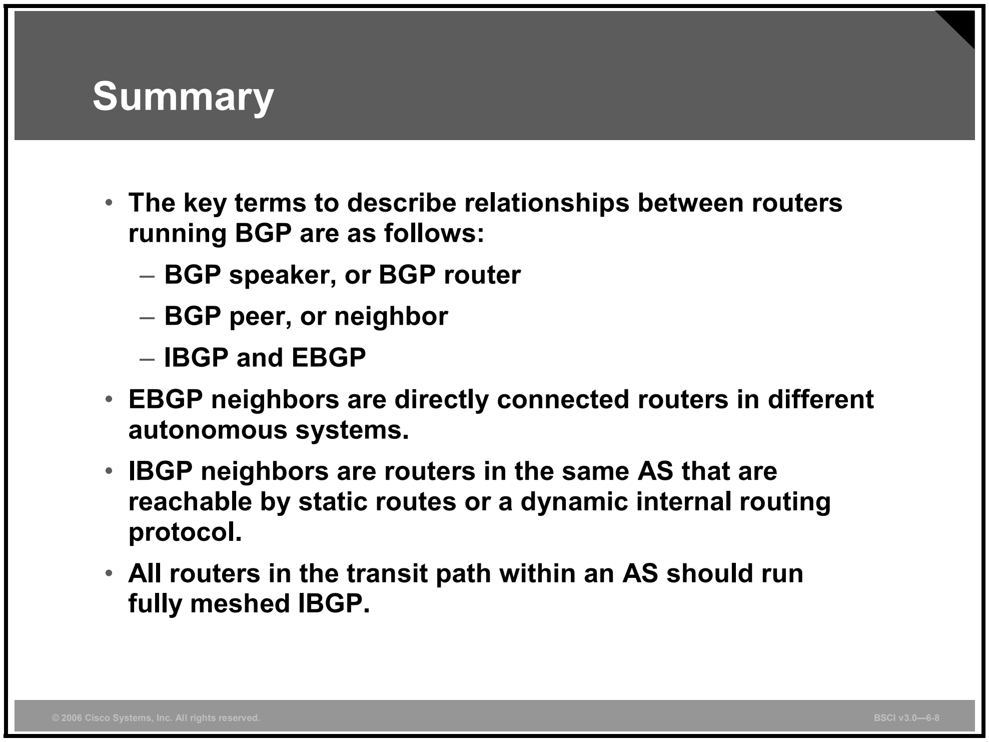
In this example, routers A, B, E, and F are the only ones running BGP. Router B has an EBGP neighbor statement for router A and an IBGP neighbor statement for router E. Router E has an EBGP neighbor statement for router F and an IBGP neighbor statement for router B. Routers C and D are not running BGP. Routers B, C, D, and E are running OSPF as their IGP.

Network 10.0.0.0 is owned by AS 65101 and is advertised to router B via an EBGP session. Router B advertises it to router E through an IBGP session. Routers C and D never learn about this network, because it is not redistributed into the local routing protocol (OSPF), and routers C and D are not running BGP. If router E advertises this network to router F in AS 65103, and router F starts forwarding packets to network 10.0.0.0 through AS 65102, router E would send the packets to its BGP peer, router B. However, to get to router B, the packets must go through router C or D, but those routers do not have an entry in their routing tables for network 10.0.0.0. Thus, when router E forwards packets with a destination address in network 10.0.0.0 to either routers C or D, those routers discard the packets.

Even if routers C and D have a default route going to the exit points of the AS (routers B and E), there is a good chance that when router E sends a packet for network 10.0.0.0 to routers C or D, those routers may send it back to router E, which will forward it again to routers C or D, causing a routing loop. To solve this problem, BGP must be implemented on routers C and D. In other words, all routers in the transit path within the AS must be running BGP; the IBGP sessions must be fully meshed.

Summary

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



Summary

- The key terms to describe relationships between routers running BGP are as follows:
  - BGP speaker, or BGP router
  - BGP peer, or neighbor
  - IBGP and EBGP
- EBGP neighbors are directly connected routers in different autonomous systems.
- IBGP neighbors are routers in the same AS that are reachable by static routes or a dynamic internal routing protocol.
- All routers in the transit path within an AS should run fully meshed IBGP.

Lesson 3

Configuring Basic BGP Operations

Overview

This lesson presents the commands to properly configure Border Gateway Protocol (BGP) so that it will do the following:

- Establish a neighbor relationship
- Set the next-hop address
- Set the source IP address of a BGP update
- Announce networks to other BGP routers

This lesson also presents the various neighbor states through which BGP progresses to establish a BGP session and offers tips for troubleshooting BGP if the session is stuck in the active or idle state. BGP authentication between neighbors is examined. This lesson also shows how to use the **show** and **debug** commands for troubleshooting BGP. You must have a thorough understanding of this material to use BGP.

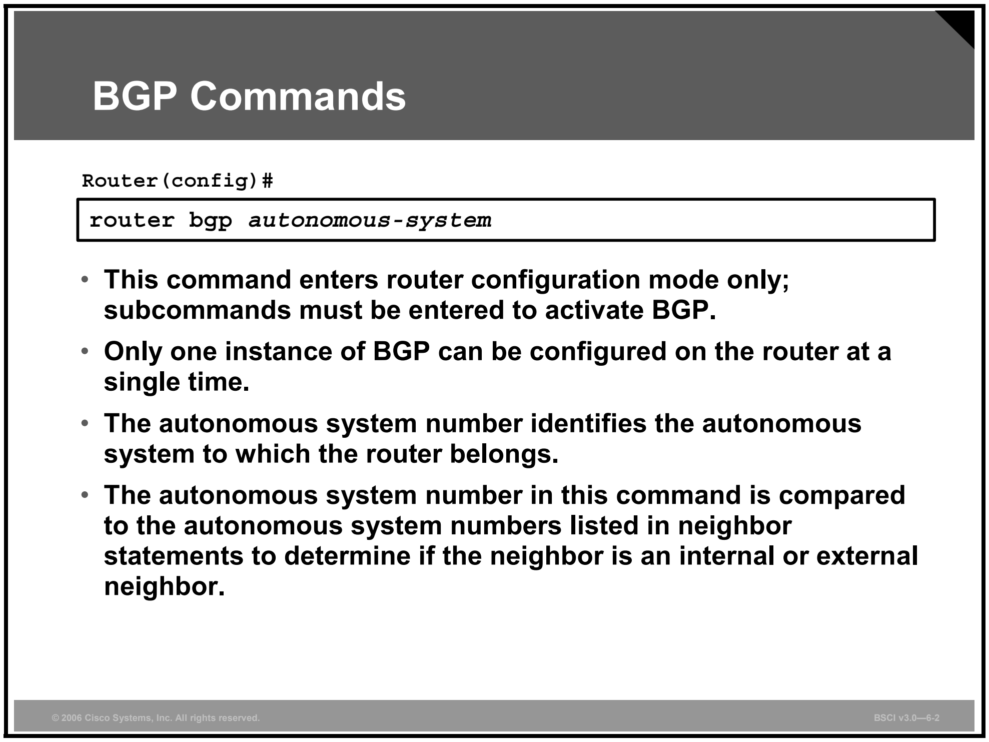
Objectives

Upon completing this lesson, you will be able to implement BGP operation. This ability includes being able to meet these objectives:

- Describe how to initiate basic BGP configuration
- Describe how to activate a BGP session for external and internal neighboring routers
- Describe how to administratively shut down and re-enable a BGP neighbor
- Describe what needs to be considered when configuring BGP
- Describe BGP neighbor states
- Describe how to configure MD5 authentication on the BGP TCP connection between two routers
- Describe how to troubleshoot BGP

Initiate Basic BGP Configuration

This topic describes how to configure basic BGP operations.



BGP Commands

```
Router(config)#
router bgp autonomous-system
```

- **This command enters router configuration mode only; subcommands must be entered to activate BGP.**
- **Only one instance of BGP can be configured on the router at a single time.**
- **The autonomous system number identifies the autonomous system to which the router belongs.**
- **The autonomous system number in this command is compared to the autonomous system numbers listed in neighbor statements to determine if the neighbor is an internal or external neighbor.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-2

The syntax of basic BGP configuration commands is similar to the syntax for configuring internal routing protocols. However, there are significant differences in how BGP functions.

Use the **router bgp** *autonomous-system* command to identify to the router that any subsequent subcommands belong to this routing process. This command also identifies the local autonomous system (AS) in which this router belongs. The router needs to be informed of the AS so it can determine whether the BGP neighbors to be configured next are Internal Border Gateway Protocol (IBGP) or External Border Gateway Protocol (EBGP) neighbors.

The router bgp Command Parameter

The table describes the syntax of the **router bgp** command.

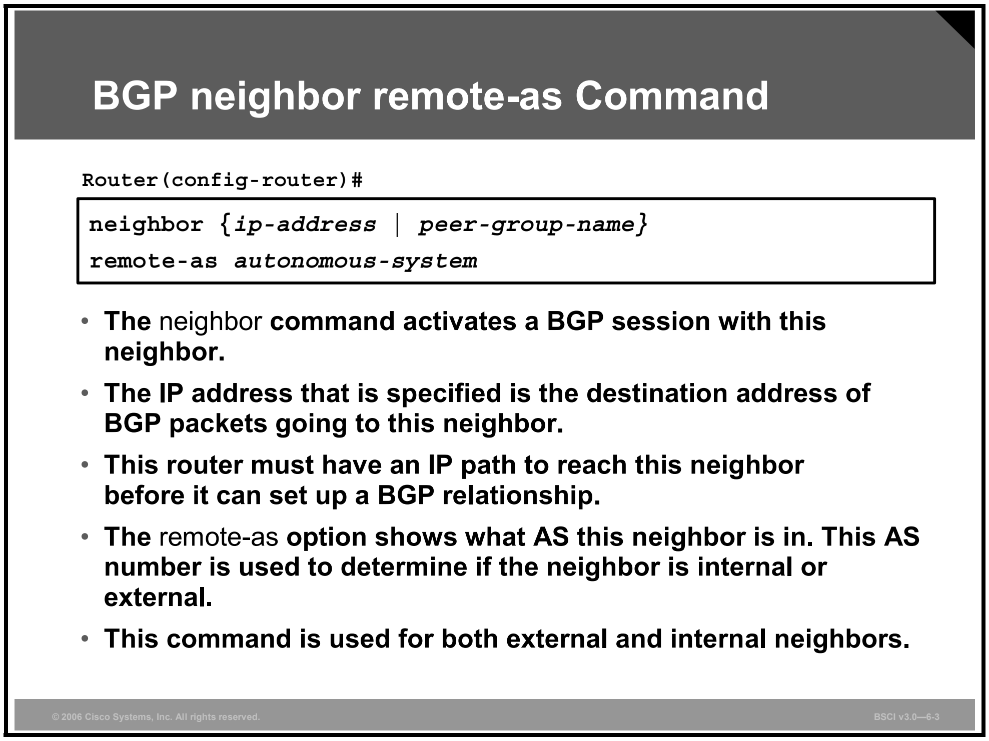
Parameter	Description
autonomous-system	Identifies the local AS number

The **route bgp** command alone cannot activate BGP on a router. You must enter at least one subcommand under the **router bgp** command to activate the BGP process on the router.

If you place your router in AS A and then try to configure a new **router bgp** B command, the router informs you that you are currently configured for AS A. You must insert the AS number in the **router bgp** command so that the router can properly identify the relationship between the neighboring router and itself.

Activate a BGP Session

This topic describes how to activate a BGP session for external and internal neighboring routers.



BGP neighbor remote-as Command

```
Router(config-router)#
neighbor {ip-address | peer-group-name}
remote-as autonomous-system
```

- **The** neighbor **command activates a BGP session with this neighbor.**
- **The IP address that is specified is the destination address of BGP packets going to this neighbor.**
- **This router must have an IP path to reach this neighbor before it can set up a BGP relationship.**
- **The** remote-as **option shows what AS this neighbor is in. This AS number is used to determine if the neighbor is internal or external.**
- **This command is used for both external and internal neighbors.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-3

You use the **neighbor** *ip-address* **remote-as** *autonomous-system* command to activate a BGP session for external and internal neighboring routers.

This command identifies a peer router with which the local router will establish a session.

The neighbor remote-as Command Parameters

This table describes the syntax of the **neighbor remote-as** command.

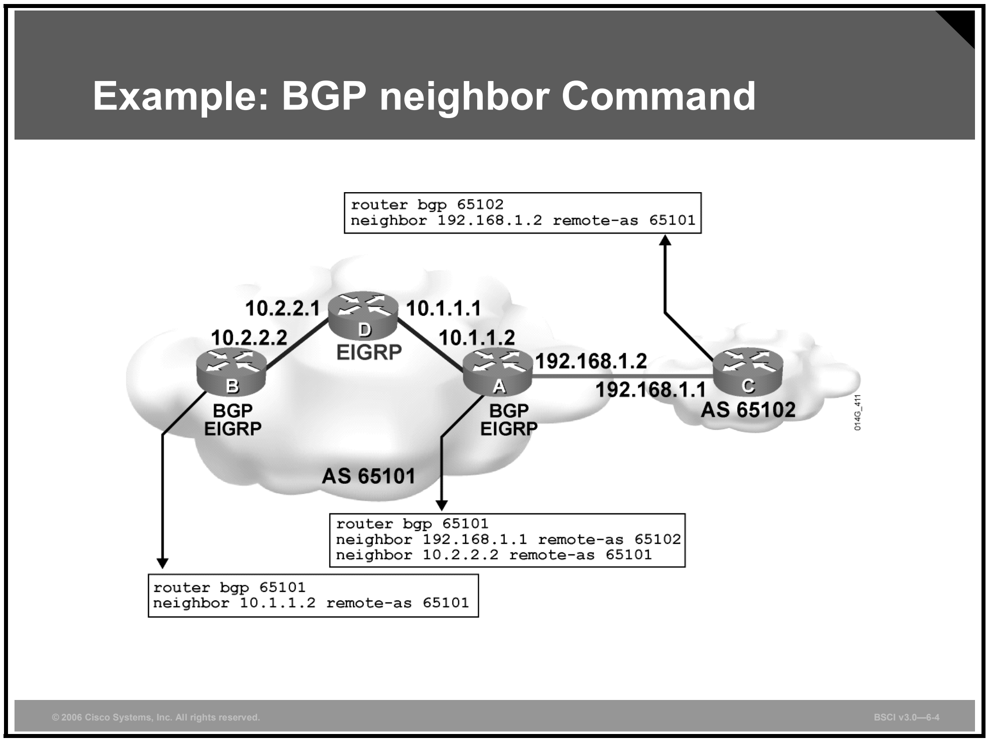
Parameter	Description
ip-address	Identifies the peer router
peer-group-name	Identifies the name of a BGP peer group
autonomous-system	Identifies the AS of the peer router

Note A peer group is a group of BGP neighbors of the router being configured that all have the same update policies. Peer groups are described later in this lesson.

This command is mandatory for the establishment of each neighboring router relationship.

The address that is used in this command is the destination address for all BGP packets going to this neighboring router. For BGP to pass BGP routing information, this address must be reachable, because BGP attempts to establish a TCP session and exchange BGP updates with the device at this IP address.

The AS number that is a part of this command is used to identify whether this neighbor is an EBGP neighbor or an IBGP neighbor. If the AS number is the same as the AS number for this router, that neighbor is an IBGP neighbor and the IP address listed in this **neighbor** command does not have to be directly connected. If the AS number is different from the AS number for this router, this neighbor is an EBGP neighbor and the address in this **neighbor** command must be directly connected by default.



Example: BGP neighbor Command

```
router bgp 65102
neighbor 192.168.1.2 remote-as 65101
```

```
router bgp 65101
neighbor 192.168.1.1 remote-as 65102
neighbor 10.2.2.2 remote-as 65101
```

```
router bgp 65101
neighbor 10.1.1.2 remote-as 65101
```

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—6-4

Example: BGP neighbor Command

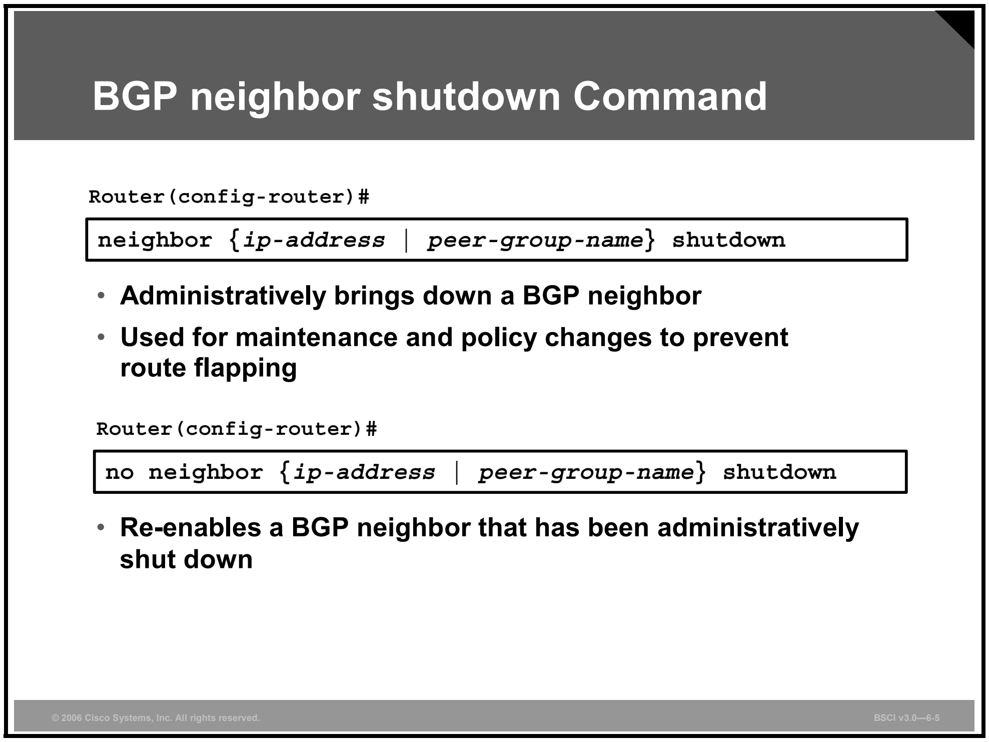
In this figure, router A in AS 65101 has two neighbor statements. Router A knows that router C (neighbor 192.168.1.1 remote-as 65102) is an external neighbor because AS 65102 in the neighbor statement for router C does not match the AS number of router A, which is AS 65101. Router A can reach AS 65102 via 192.168.1.1, which is directly connected to router A.

Neighbor 10.2.2.2 (router B) is in the same AS as router A; the second neighbor statement on router A defines router B as an IBGP neighbor.

AS 65101 runs Enhanced Interior Gateway Routing Protocol (EIGRP) between all internal routers. Router A has an EIGRP path to reach IP address 10.2.2.2. As an IBGP neighbor, router B can be multiple routers away from router A.

Shutting Down a BGP Neighbor

This topic describes how to administratively shut down and re-enable a BGP neighbor.



BGP neighbor shutdown Command

```
Router(config-router)#
neighbor {ip-address | peer-group-name} shutdown
```

- **Administratively brings down a BGP neighbor**
- **Used for maintenance and policy changes to prevent route flapping**

```
Router(config-router)#
no neighbor {ip-address | peer-group-name} shutdown
```

- **Re-enables a BGP neighbor that has been administratively shut down**

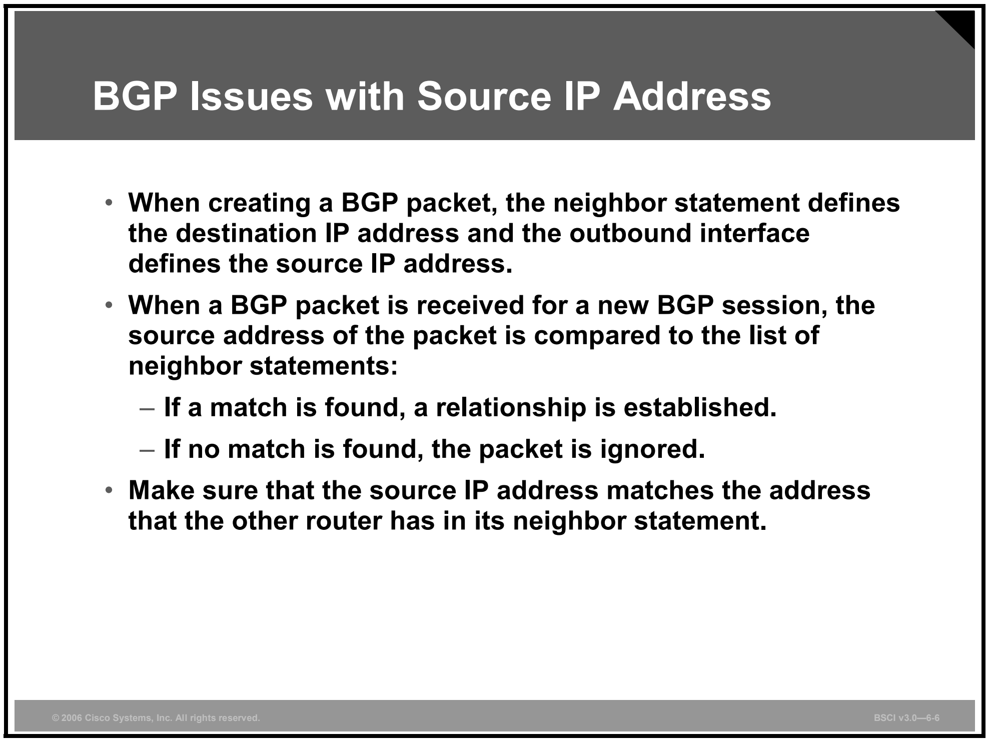
© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-5

Use the **neighbor** *ip-address* **shutdown** commands to administratively shut down and re-enable a BGP neighbor.

If you implement major policy changes to a neighboring router and you change multiple parameters, you must administratively shut down the neighboring router, implement the changes, and then bring the neighboring router back up with the **no neighbor** *ip-address* **shutdown** command.

BGP Configuration Considerations

This topic describes what needs to be considered when configuring BGP.



BGP Issues with Source IP Address

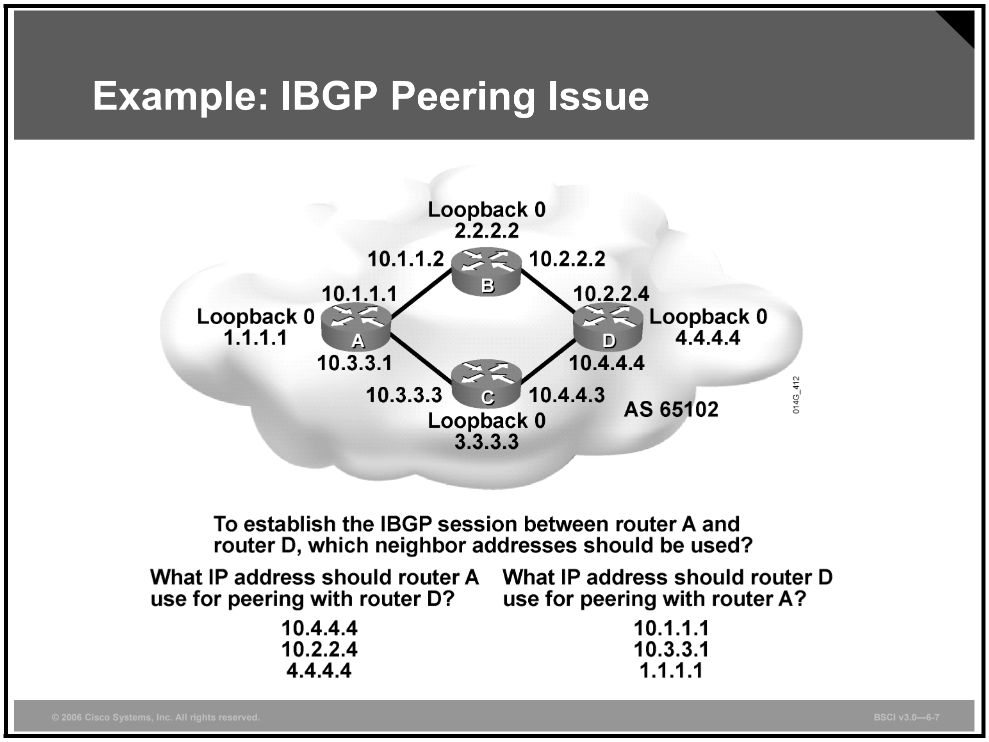
- **When creating a BGP packet, the neighbor statement defines the destination IP address and the outbound interface defines the source IP address.**
- **When a BGP packet is received for a new BGP session, the source address of the packet is compared to the list of neighbor statements:**
  - **If a match is found, a relationship is established.**
  - **If no match is found, the packet is ignored.**
- **Make sure that the source IP address matches the address that the other router has in its neighbor statement.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-6

The BGP neighbor statement informs the router of the destination IP address for each update packet. The router must decide which IP address to use as the source IP address in the BGP routing update.

When a router creates a BGP packet for a neighbor, it checks the routing table for the destination network to reach that neighbor. The IP address of the outbound interface, as the routing table indicates, is used as the source IP address of the BGP packet.

This source IP address must match the address in the corresponding neighbor statement on the other router. Otherwise, the routers will not be BGP peers because they are not able to establish the BGP session.



Example: IBGP Peering Issue

To establish the IBGP session between router A and router D, which neighbor addresses should be used?

What IP address should router A use for peering with router D?	What IP address should router D use for peering with router A?
10.4.4.4	10.1.1.1
10.2.2.4	10.3.3.1
4.4.4.4	1.1.1.1

BSCI v3.0—6-7

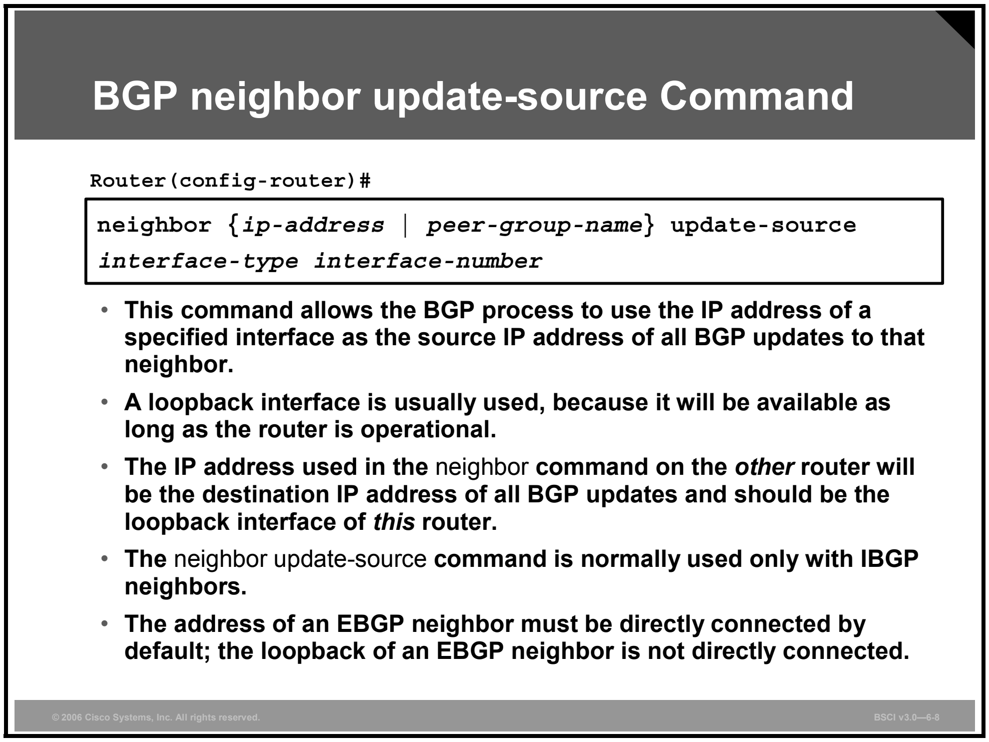
Example: IBGP Peering Issue

To establish the IBGP session between router A and router D, as shown in this figure, which neighbor IP address should be used?

The problem is as follows: If router D uses **neighbor 10.3.3.1 remote-as 65102**, but router A is sending the BGP packets to router D via router B, the source IP address will be 10.1.1.1.

When router D receives this BGP packet via router B, it will not recognize this BGP packet because 10.1.1.1 was not configured as a neighbor of router D; therefore, the IBGP session between router A and router D cannot be established.

A solution to this problem is to establish the IBGP session using a loopback interface when there are multiple paths between the IBGP neighbors.



BGP neighbor update-source Command

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name} update-source
interface-type interface-number
```

- **This command allows the BGP process to use the IP address of a specified interface as the source IP address of all BGP updates to that neighbor.**
- **A loopback interface is usually used, because it will be available as long as the router is operational.**
- **The IP address used in the** neighbor **command on the *other* router will be the destination IP address of all BGP updates and should be the loopback interface of *this* router.**
- **The** neighbor update-source **command is normally used only with IBGP neighbors.**
- **The address of an EBGP neighbor must be directly connected by default; the loopback of an EBGP neighbor is not directly connected.**

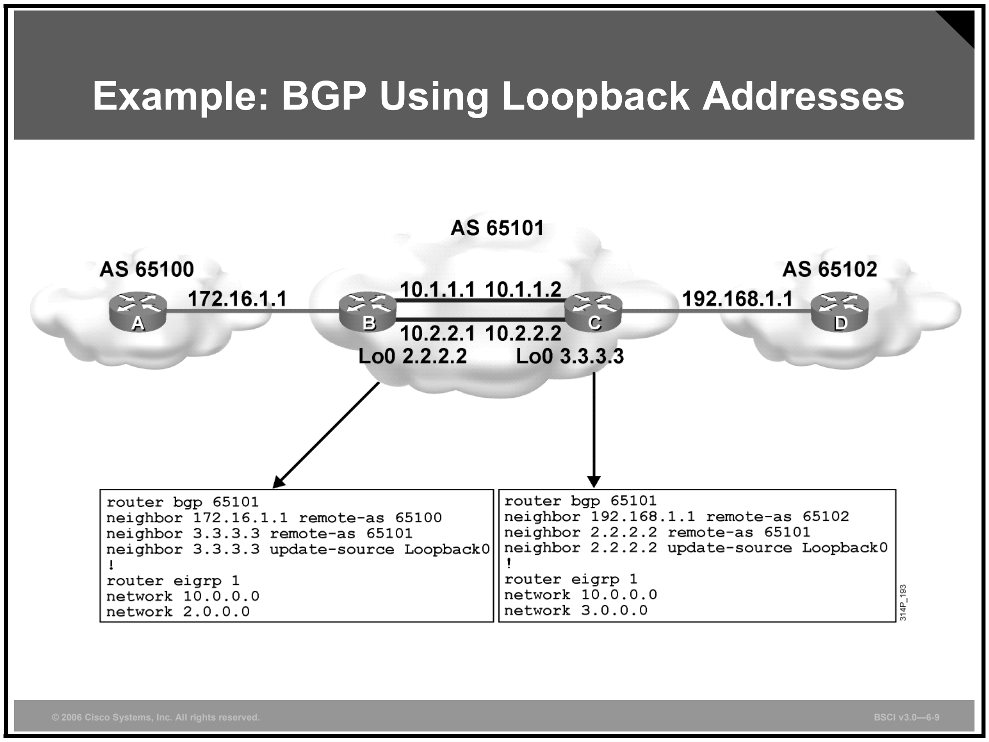
The **update-source** option in the **neighbor** command overrides the default source IP address used for BGP packets. It is necessary to tell the router which IP address to use as the source address for all BGP packets if you want to use a loopback interface instead of the physical interface.

If you do not use the **update-source** option in the **neighbor** command, an announcement going to a neighbor uses the IP address of the exiting interface as the source address for a packet.

When a router creates a packet, whether it is a routing update, a ping, or any other type of IP packet, the router does a lookup in the routing table for the destination address. The routing table lists the appropriate interface to get to the destination address. The address of this outbound interface is used as the source address of that packet by default.

Consider what would happen if a neighboring router uses the loopback interface address in its **neighbor** command for this router, but you do not use the **neighbor update-source** command on this router. When the neighboring router receives an update packet and looks at the source address of the packet, it sees that it has no neighbor relationship with that source address, so it discards the packet.

BGP does not accept unsolicited updates; it must be aware of every neighboring router and have a neighbor statement for it.



Example: BGP Using Loopback Addresses

```
router bgp 65101
neighbor 172.16.1.1 remote-as 65100
neighbor 3.3.3.3 remote-as 65101
neighbor 3.3.3.3 update-source Loopback0
!
router eigrp 1
network 10.0.0.0
network 2.0.0.0
```

```
router bgp 65101
neighbor 192.168.1.1 remote-as 65102
neighbor 2.2.2.2 remote-as 65101
neighbor 2.2.2.2 update-source Loopback0
!
router eigrp 1
network 10.0.0.0
network 3.0.0.0
```

Multiple paths can exist to reach each neighbor when you peer with IBGP neighboring routers. If the BGP router is using a neighbor address that is assigned to a specific interface on another router, and that interface goes down, the router pointing to this address loses its BGP session with that neighbor.

If the router peers instead with the loopback interface of the other router, the loopback interface will always be available as long as the router itself does not fail. This peering arrangement adds resiliency to the IBGP sessions because the routers are not tied into a physical interface, which may go down for any number of reasons.

To peer with the loopback of another internal neighbor, the first router would point the neighbor statement at the loopback address of the other internal neighbor. Ensure that both routers have a route to the loopback address of the other neighbor in their routing table. Also ensure that both routers are announcing their loopback addresses into their local routing protocol.

Example: BGP Using Loopback Addresses

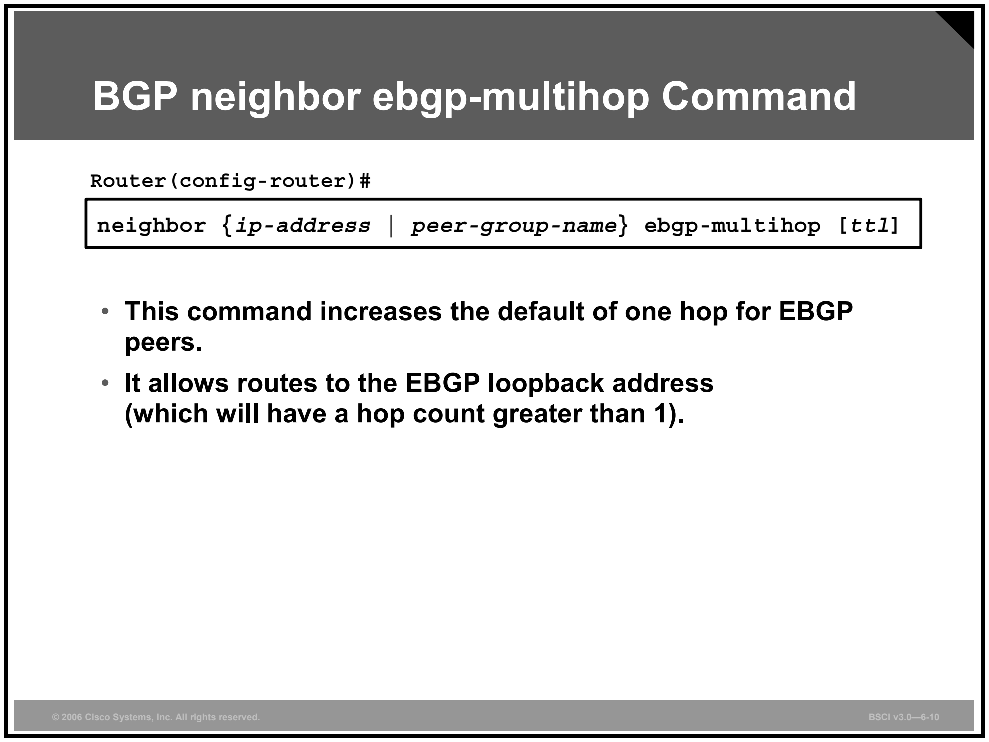
In this figure, router B has router A as an EBGP neighbor. The only reachable address for router B to use for a neighbor address in BGP is the directly connected address of 172.16.1.1. Router B has multiple paths to reach router C, an IBGP neighbor.

All networks, including the IP network for the loopback interface of router C, can be reached from router B. Router B can reach these networks because routers B and C exchange EIGRP updates; router B and router A do not exchange EIGRP updates.

The neighbor relationship between routers B and C is not tied to a physical interface because router B peers with the loopback interface on router C and uses its loopback address as the source IP address, and vice versa. If router B instead peered with 10.1.1.2 on router C and that interface went down, the BGP neighbor relationship would also go down.

The **neighbor update-source** command should be used on both routers. If router B points to loopback address 3.3.3.3 of router C, and router C points at loopback address 2.2.2.2 of router B, and neither uses the **neighbor update-source** command, the BGP session between these routers will not start.

Router B would send a BGP open packet to router C with the source IP address being either 10.1.1.1 or 10.2.2.1. Router C would review the source IP address and attempt to match it against its list of known neighbors. Router C would not find a match and would not respond to the open message from router B.



BGP neighbor ebgp-multihop Command

```
Router(config-router)#
neighbor {ip-address | peer-group-name} ebgp-multihop [ttl]
```

- **This command increases the default of one hop for EBGP peers.**
- **It allows routes to the EBGP loopback address (which will have a hop count greater than 1).**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-10

When an EBGP router is peering with an external neighbor, the only address that it can reach without further configuration is the interface that is directly connected to that EBGP router. Because internal routing information is not exchanged with external peers, the router has to point to a directly connected address for that external neighbor.

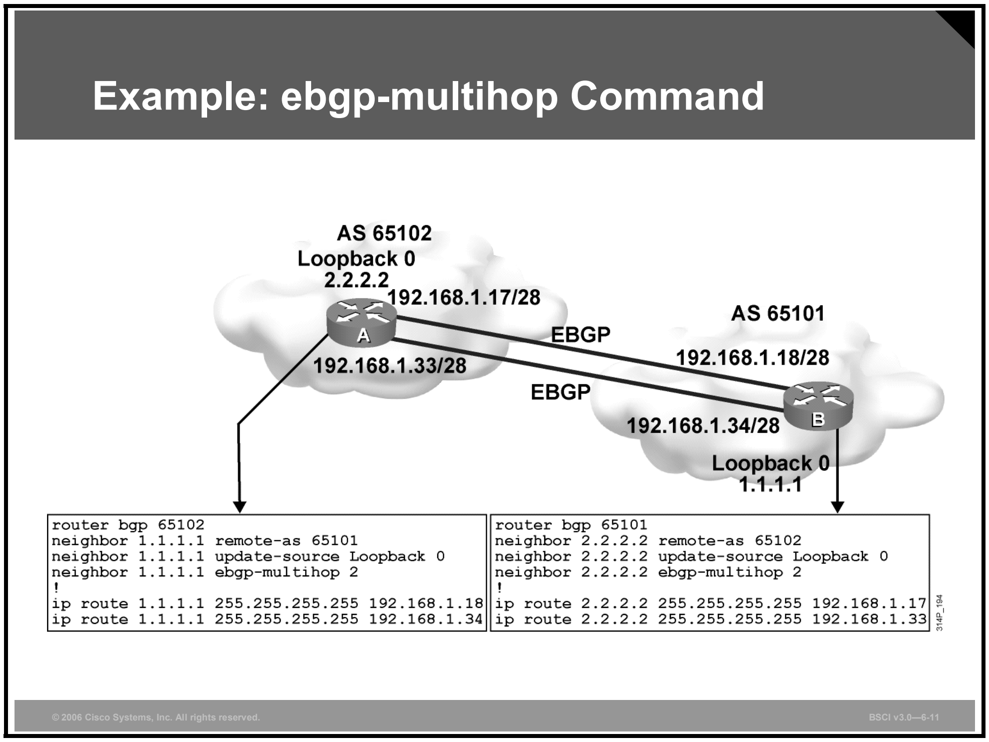
A loopback interface is never directly connected. Therefore, if you want to use a loopback interface instead, use static routes pointing at the physical address of the directly connected network (the next-hop address). In addition to the static route, you need to use the **neighbor** *ip-address* **ebgp-multihop** [*ttl*] router configuration command.

This command allows the router to accept and attempt BGP connections to external peers residing on networks that are not directly connected. This command increases the default of one hop for EBGP peers by changing the default Time to Live (TTL) value of 1. It allows routes to the EBGP loopback address with a hop value greater than 1. By default, the TTL is set to 255 with this command. This command is of value when redundant paths exist between EBGP neighbors.

The neighbor ebgp multihop Command Parameters

The table describes the syntax of the **neighbor ebgp multihop** command.

Parameter	Description
ip-address	IP address of the BGP-speaking neighbor
peer-group-name	Name of a BGP peer group
ttl	(Optional) TTL in the range from 1 to 255 hops



Example: ebgp-multihop Command

```
router bgp 65102
neighbor 1.1.1.1 remote-as 65101
neighbor 1.1.1.1 update-source Loopback 0
neighbor 1.1.1.1 ebgp-multihop 2
!
ip route 1.1.1.1 255.255.255.255 192.168.1.18
ip route 1.1.1.1 255.255.255.255 192.168.1.34
```

```
router bgp 65101
neighbor 2.2.2.2 remote-as 65102
neighbor 2.2.2.2 update-source Loopback 0
neighbor 2.2.2.2 ebgp-multihop 2
!
ip route 2.2.2.2 255.255.255.255 192.168.1.17
ip route 2.2.2.2 255.255.255.255 192.168.1.33
```

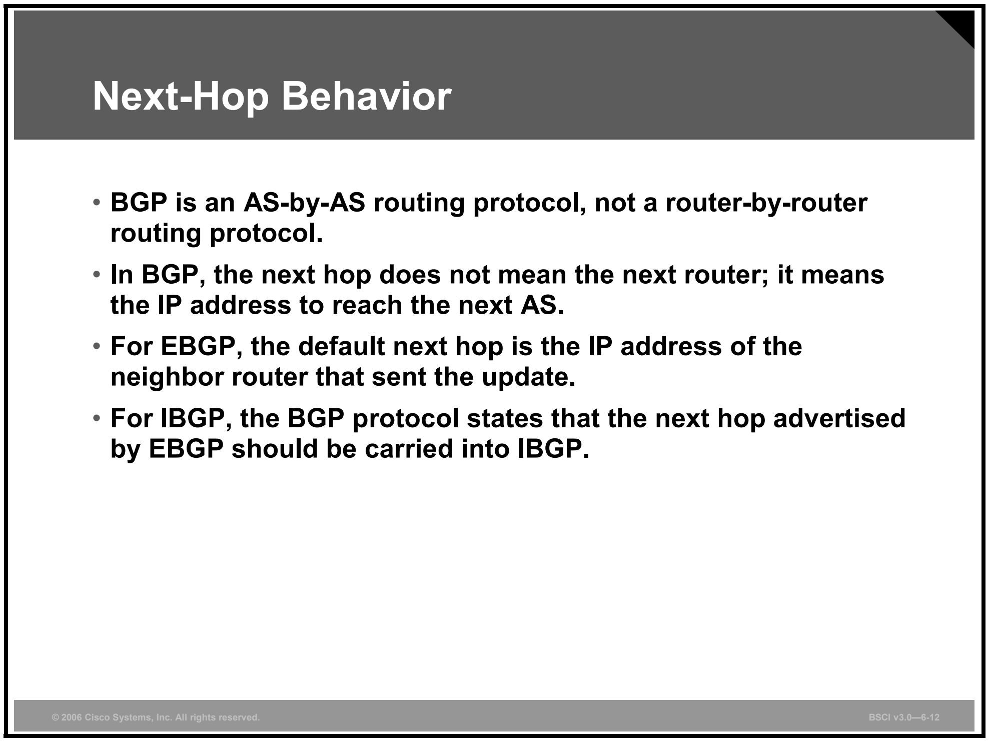
Example: ebgp-multihop Command

In the figure, router A in AS 65102 has two paths to router B in AS 65101. If router A uses a single neighbor statement and points at 192.168.1.18 on router B of AS 65101 and that link goes down, there will be no BGP session between these autonomous systems and no packets would pass from one AS to the next, although another link exists. If router A instead uses two neighbor statements pointing at 192.168.1.18 and 192.168.1.34 on router B, it partially solves the problem. However, every BGP update that router A receives is sent to router B twice because there are two neighbor statements.

As shown in the figure, router A instead points to the loopback address of router B and vice versa, and each router uses its loopback address as the source IP address for its BGP updates. Because an interior gateway protocol (IGP) is not used between autonomous systems, neither router can reach the loopback of the other router without assistance.

Each router needs to use two static routes to inform BGP of the paths available to reach the loopback address of the other router. An EBGP neighbor address must be directly connected by default, so you must use the **neighbor ebgp-multihop** command to change the default setting of BGP and inform BGP that this neighbor IP address is more than one hop away. In the figure, the command used on router A informs BGP that the neighbor address of 1.1.1.1 is two hops away.

Note BGP is not designed to perform load balancing; paths are chosen because of policy, not based on bandwidth. BGP will choose only a single best path. Using the loopback addresses and the **neighbor ebgp-multihop** command as shown in this example allows load balancing, as well as redundancy, across the two paths between the autonomous systems.



Next-Hop Behavior

- BGP is an AS-by-AS routing protocol, not a router-by-router routing protocol.
- In BGP, the next hop does not mean the next router; it means the IP address to reach the next AS.
- For EBGP, the default next hop is the IP address of the neighbor router that sent the update.
- For IBGP, the BGP protocol states that the next hop advertised by EBGP should be carried into IBGP.

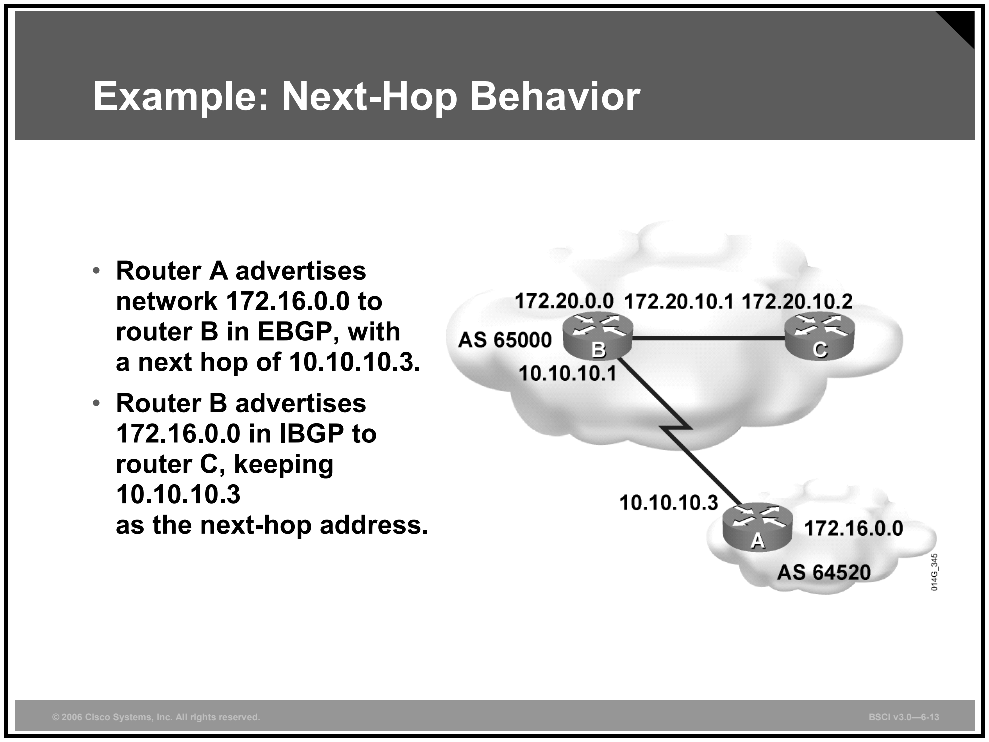
BSCI v3.0—6-12

The way in which BGP establishes an IBGP relationship is very different from the way that IGPs behave. The method that BGP uses to denote its next-hop address is also very different from the way that an IGP performs the same function.

BGP is an external routing protocol that informs the next AS about paths to other autonomous systems and the networks that those other autonomous systems own. BGP, like IGPs, is a hop-by-hop routing protocol. However, unlike IGPs, BGP routes from AS to AS, and the default next hop is the next AS. An IBGP neighboring router that learns about a network outside of its autonomous systems sees, as the next-hop address, the entry point for the next autonomous systems along the path to reach the distant network.

For EBGP, the default next hop is the IP address of the neighboring router that sent the update.

For IBGP, the BGP protocol states that the next hop advertised by EBGP should be carried into IBGP.



Example: Next-Hop Behavior

- **Router A advertises network 172.16.0.0 to router B in EBGP, with a next hop of 10.10.10.3.**
- **Router B advertises 172.16.0.0 in IBGP to router C, keeping 10.10.10.3 as the next-hop address.**

BSCI v3.0—6-13

Example: Next-Hop Behavior

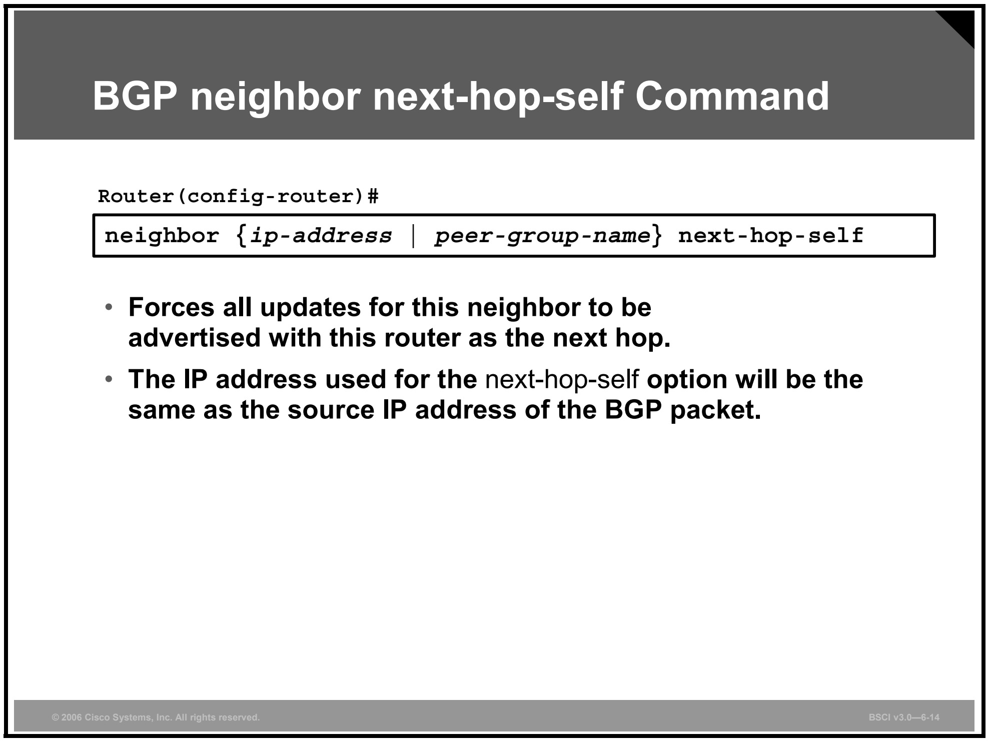
In the figure, router A advertises 172.16.0.0 to router B with a next hop of 10.10.10.3. Router B advertises 172.20.0.0 to router A with a next hop of 10.10.10.1.

For IBGP, the BGP protocol states that the next hop advertised by EBGP should be carried into IBGP. Because of this rule, router B advertises 172.16.0.0 to its IBGP peer router C with a next hop of 10.10.10.3, the address of router A. Router C knows that the next hop to reach 172.16.0.0 is 10.10.10.3, not 172.20.10.1, as you might expect.

It is therefore very important that router C knows how to reach the 10.10.10.0 subnet, either through an IGP or a static route. Otherwise, router C will drop packets destined for 172.16.0.0 because it is not able to get to the next-hop address for that network.

An IBGP neighboring router performs a recursive lookup to find out how to reach a BGP next-hop address by using its IGP entries in the routing table. For example, router C learns in a BGP update about network 172.16.0.0/16 from a route source of 172.20.10.1, router B, with a next hop of 10.10.10.3, router A. Router C installs the route to 172.16.0.0/16 in the routing table with a next hop of 10.10.10.3. Router B should announce network 10.10.10.0/24 using its IGP to router C so that router C can install that route into its routing table with a next hop of 172.20.10.1.

An IGP uses the source IP address of a routing update (route source) as the next-hop address, whereas BGP uses a separate field per network to record the next-hop address. If router C has a packet to send to 172.16.100.1, it looks up the network in the routing table and finds a BGP route with a next hop of 10.10.10.3. Because it is a BGP entry, router C completes a recursive lookup in the routing table for a path to network 10.10.10.3. The IGP has placed a route to network 10.10.10.0 in the routing table with a next hop of 172.20.10.1, so router C forwards the packet destined for 172.16.100.1 to 172.20.10.1.



BGP neighbor next-hop-self Command

```
Router(config-router)#
neighbor {ip-address | peer-group-name} next-hop-self
```

- **Forces all updates for this neighbor to be advertised with this router as the next hop.**
- **The IP address used for the** next-hop-self **option will be the same as the source IP address of the BGP packet.**

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—6-14

It is sometimes necessary to override the default next-hop behavior of a router and force it to advertise itself as the next-hop address for routes sent to a neighboring router.

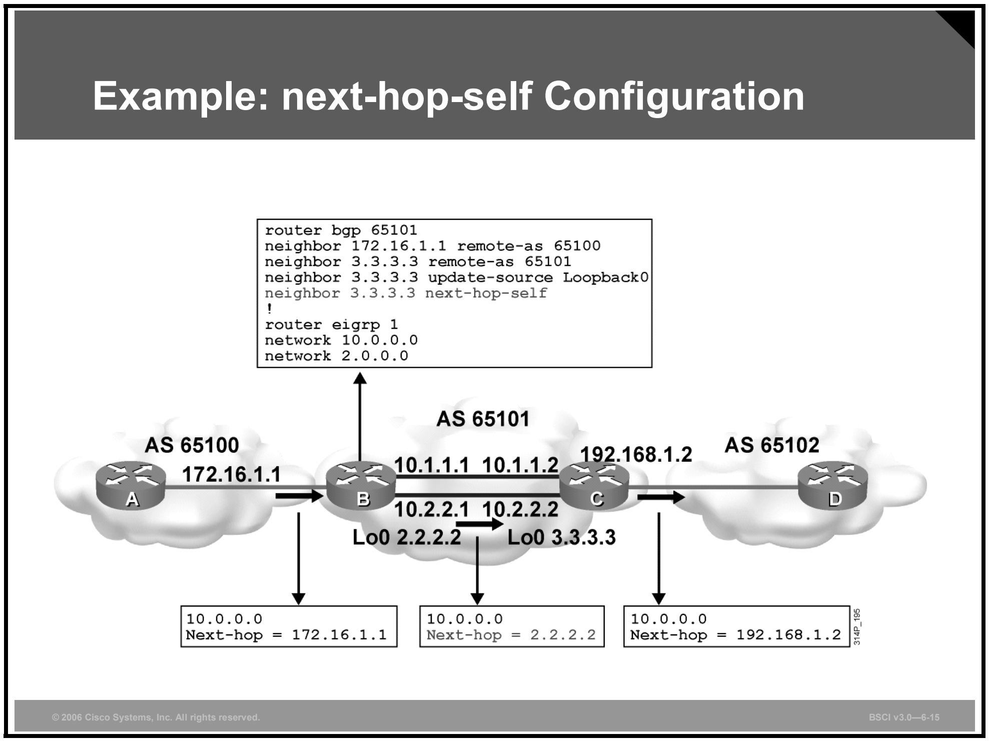
The **neighbor next-hop-self** command forces BGP to use its own IP address as the next-hop address for each network that it advertises to its IBGP neighbor, rather than letting the protocol choose the next-hop address to use.

An internal protocol, such as Routing Information Protocol (RIP), EIGRP, or Open Shortest Path First Protocol (OSPF), always uses the source IP address of a routing update as the next-hop address for each network that is placed in the routing table. The **neighbor next-hop-self** command makes BGP use the source IP address of the update as the next-hop address for each advertised network.

The neighbor next-hop-self Command Parameters

The table describes the syntax of the **neighbor next-hop-self** command.

Parameter	Description
ip-address	Identifies the peer router to which advertisements are sent with this router identified as the next hop
peer-group-name	Identifies the name of a BGP peer group



Example: next-hop-self Configuration

```
router bgp 65101
neighbor 172.16.1.1 remote-as 65100
neighbor 3.3.3.3 remote-as 65101
neighbor 3.3.3.3 update-source Loopback0
neighbor 3.3.3.3 next-hop-self
!
router eigrp 1
network 10.0.0.0
network 2.0.0.0
```

```
10.0.0.0
Next-hop = 172.16.1.1
```

```
10.0.0.0
Next-hop = 2.2.2.2
```

```
10.0.0.0
Next-hop = 192.168.1.2
```

BSCI v3.0—6-15

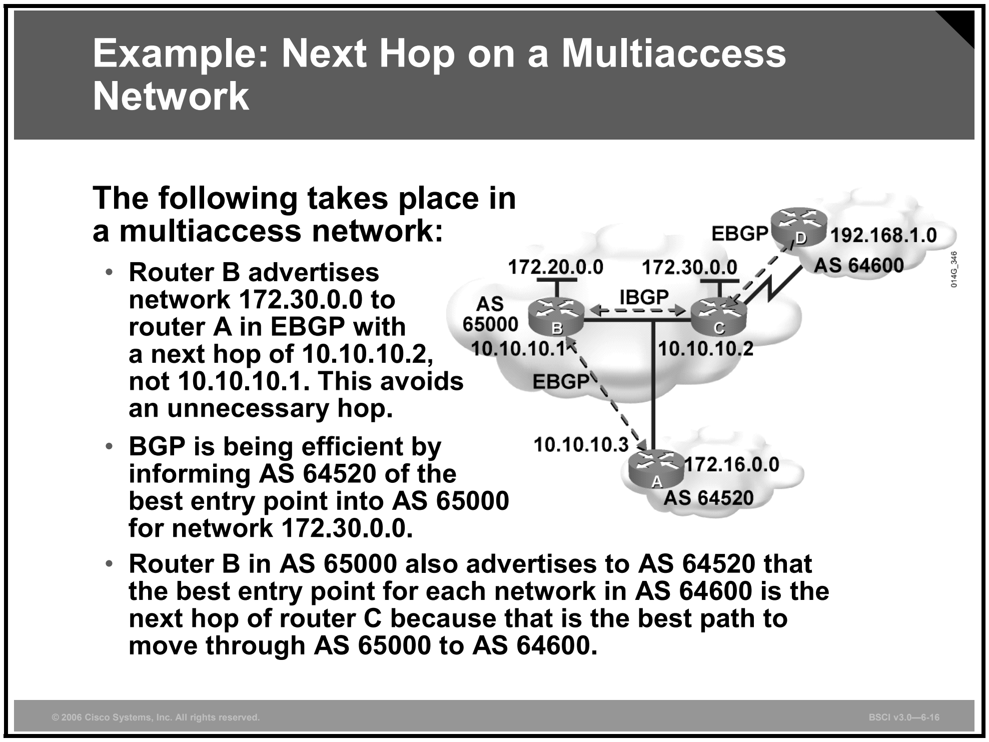
Example: next-hop-self Configuration

In this figure, router B views all routes learned from AS 65100 as having a next hop of 172.16.1.1, which is the entrance to AS 65100 for router B. When router B announces those networks to its IBGP neighbors in AS 65101, the BGP default setting is to announce that the next hop to reach each of those networks is the entrance to AS 65100 (172.16.1.1), because BGP is an AS-by-AS routing protocol.

For any BGP router to reach networks in or behind AS 65100, those routers need to reach network 172.16.1.1. Therefore, you need to include the network that represents 172.16.1.1 in the internal routing protocol.

In this example, however, router B uses the **neighbor next-hop-self** command to change the default BGP settings. Once this command is given, router B advertises a next hop of 2.2.2.2 (the IP address of the loopback interface) to its IBGP neighbor, because that is the source IP address of the routing update to its IBGP neighbor (set with the **neighbor update-source** command).

When router C announces networks that are in or behind AS 65101 to EBGP neighbors, such as router D in AS 65102, router C, by default, uses its outbound interface address 192.168.1.2 as the next-hop address. This address is also the default next-hop address for router D to use to reach any networks in or behind AS 65101.



Example: Next Hop on a Multiaccess Network

The following takes place in a multiaccess network:

- **Router B advertises network 172.30.0.0 to router A in EBGP with a next hop of 10.10.10.2, not 10.10.10.1. This avoids an unnecessary hop.**
- **BGP is being efficient by informing AS 64520 of the best entry point into AS 65000 for network 172.30.0.0.**
- **Router B in AS 65000 also advertises to AS 64520 that the best entry point for each network in AS 64600 is the next hop of router C because that is the best path to move through AS 65000 to AS 64600.**

BSCI v3.0—6-16

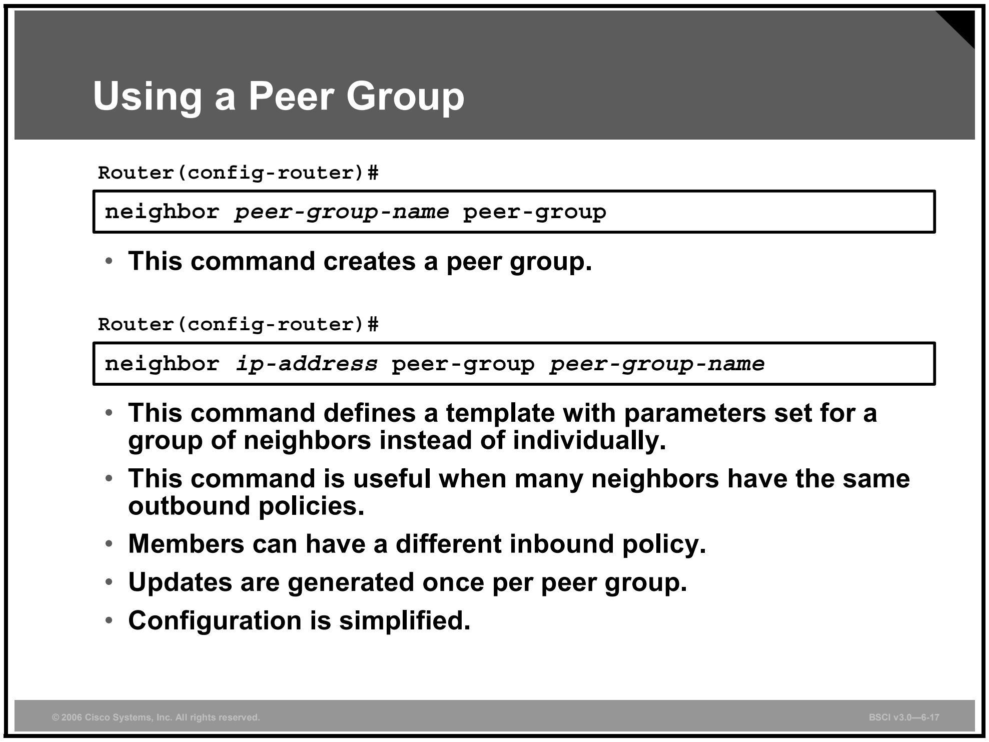
When running BGP over a multiaccess network such as Ethernet, a BGP router adjusts the next-hop address to avoid inserting additional hops into the network. This feature is sometimes called a third-party next hop.

Example: Next Hop on a Multiaccess Network

As shown in the figure, routers B and C in AS 65000 are running an IGP so that router B can reach network 172.30.0.0 via 10.10.10.2. Router B also runs EBGP with router A. When router B sends a BGP update to router A regarding 172.30.0.0, it uses 10.10.10.2 as the next hop and not its own IP address (10.10.10.1). Because the network between the three routers is a multiaccess network, router A uses router C as a next hop to reach 172.30.0.0, rather than making an extra hop via router B.

The next-hop address issue makes more sense when you review it from an Internet service provider (ISP) perspective. A large ISP at a public peering point has multiple routers peering with different neighboring routers. It is not possible for one router to peer with every neighboring router at the major public peering points. For example, in the figure, router B may peer with AS 64520, and router C may peer with AS 64600.

From the perspective of router A, it must have a path through AS 65000 to get to networks in and behind AS 64600. Router A has a neighbor relationship with only router B in AS 65000; however, router B does not handle traffic going to AS 64600. The preferred path of router B to AS 64600 is through router C, 10.10.10.2. Router B must advertise the networks for AS 64600 to router A, 10.10.10.3. Router B notices that routers A and C are on the same subnet, so router B informs router A to install the AS 64600 networks with a next hop of 10.10.10.2 and not 10.10.10.1.



Using a Peer Group

```
Router(config-router)#
neighbor peer-group-name peer-group
```

- **This command creates a peer group.**

```
Router(config-router)#
neighbor ip-address peer-group peer-group-name
```

- **This command defines a template with parameters set for a group of neighbors instead of individually.**
- **This command is useful when many neighbors have the same outbound policies.**
- **Members can have a different inbound policy.**
- **Updates are generated once per peer group.**
- **Configuration is simplified.**

BSCI v3.0—6-17

In BGP, neighboring routers are often configured with the same update policies. For example, the neighboring routers may have the same filtering applied. On Cisco Systems routers, neighboring routers with the same update policies can be grouped into peer groups to simplify configuration and to make updating more efficient and improve performance. Members of the peer group inherit all the configuration options of the peer group. You can configure the router to override these options for some members if these options affect inbound advertisements but not outbound updates.

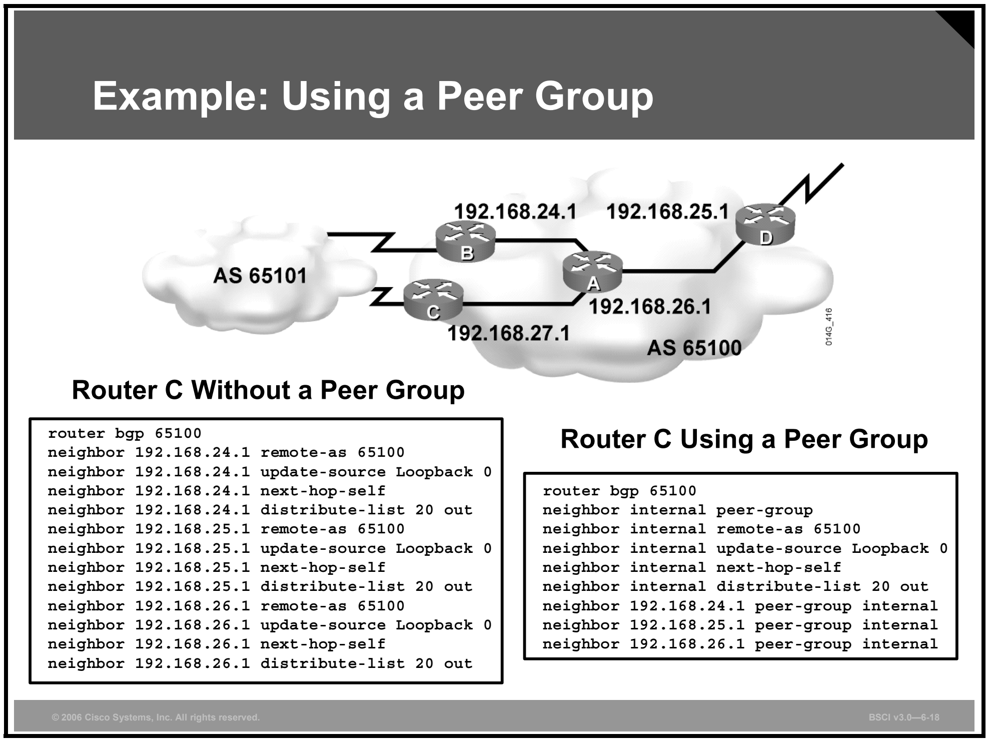
Peer groups are more efficient because updates are generated only once per peer group rather than repetitiously for each neighboring router. The generated update is replicated for each neighbor that is part of the internal peer group. Peer groups save processing time in generating the updates for all IBGP neighbors. The peer group name is local to the router on which it is configured, and it is not passed to any other router. Peer groups make the router configuration easier to read and manage.

Use the following command to create a peer group and define the name for linking similar neighboring routers together:

```
neighbor peer-group-name peer-group
```

Use the second command in the figure, **neighbor** *ip-address* **peer-group** *peer-group-name*, to link the address of a neighboring router to a specific peer group name. A neighboring router can be part of only one peer group. This command allows you to enter the peer group name instead of entering the IP address in other commands, for example, to link a policy to the group of neighboring routers. You must enter the **neighbor** *peer-group-name* **peer-group** command before the router will accept the second command.

Note Recent releases of Cisco IOS software contain a BGP Dynamic Update Peer-Groups using Peer Templates feature to dynamically optimize update groups of neighbors for shared outbound policies. More information on this feature can be found on Cisco.com.



Example: Using a Peer Group

Router C Without a Peer Group

```
router bgp 65100
neighbor 192.168.24.1 remote-as 65100
neighbor 192.168.24.1 update-source Loopback 0
neighbor 192.168.24.1 next-hop-self
neighbor 192.168.24.1 distribute-list 20 out
neighbor 192.168.25.1 remote-as 65100
neighbor 192.168.25.1 update-source Loopback 0
neighbor 192.168.25.1 next-hop-self
neighbor 192.168.25.1 distribute-list 20 out
neighbor 192.168.26.1 remote-as 65100
neighbor 192.168.26.1 update-source Loopback 0
neighbor 192.168.26.1 next-hop-self
neighbor 192.168.26.1 distribute-list 20 out
```

Router C Using a Peer Group

```
router bgp 65100
neighbor internal peer-group
neighbor internal remote-as 65100
neighbor internal update-source Loopback 0
neighbor internal next-hop-self
neighbor internal distribute-list 20 out
neighbor 192.168.24.1 peer-group internal
neighbor 192.168.25.1 peer-group internal
neighbor 192.168.26.1 peer-group internal
```

Example: Using a Peer Group

In this figure, AS 65100 has four routers running IBGP. All these IBGP neighbors are peering with each other using the loopback 0 interface of each and are using the IP address of their loopback 0 interface as the source IP address for all BGP packets. Each router is using one of its own IP addresses as the next-hop address for each network advertised through BGP. These are outbound policies.

In addition, router C has an outbound distribution list associated with each IBGP neighbor. This outbound filter performs the same function as the **distribute-list** command that you use for internal routing protocols; however, it is linked to a specific neighbor for use with BGP. The ISP behind router C may be announcing RFC 1918 private address space to router C. Router C does not want to pass these networks to other routers running BGP in AS 65100.

To accomplish this goal, access list 20 might look like the following:

```
access-list 20 deny 10.0.0.0 0.255.255.255
access-list 20 deny 172.16.0.0 0.31.255.255
access-list 20 deny 192.168.0.0 0.0.255.255
access-list 20 permit any
```

The figure shows the configuration on router C when the router is not using a peer group. All IBGP neighbors have the outbound distribute list link to them individually. If router C receives a change from AS 65101, router C must generate an individual update for each IBGP neighbor and run each update against distribute list 20. If router C has a large number of IBGP neighbors, the processing power needed to inform the IBGP neighbors of the changes in AS 65101 could be extensive.

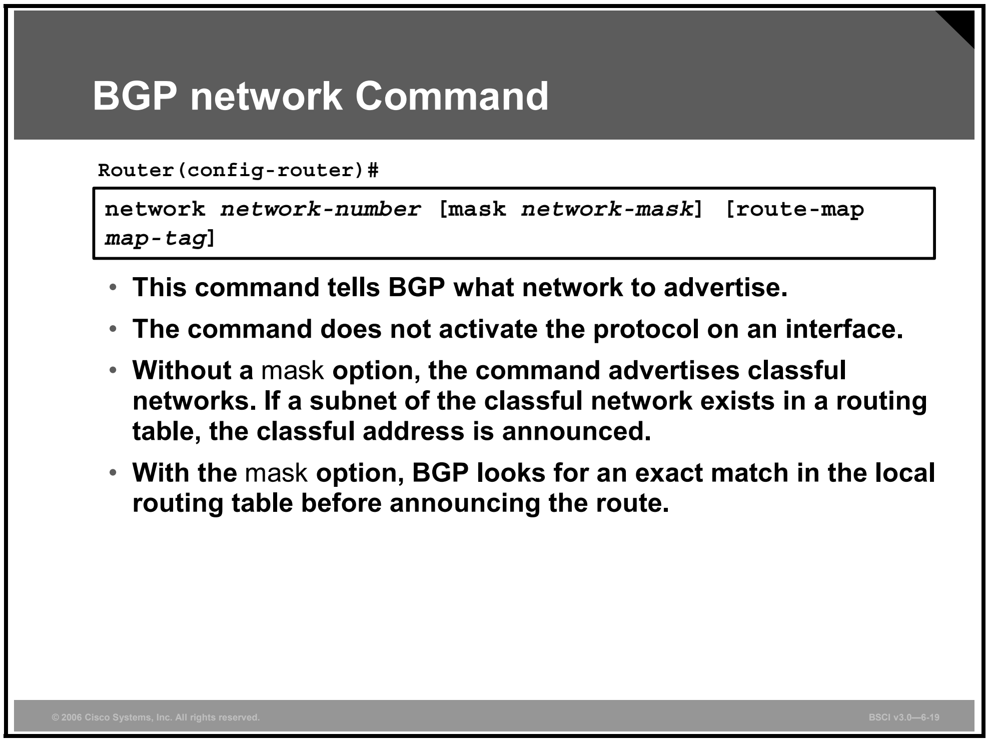
The figure also shows the configuration on router C when it is using a peer group called "internal." The **update-source**, **next-hop-self**, and **distribute-list 20 out** commands are all linked to peer group internal, which in turn is linked to each of the IBGP neighbors.

If router C receives a change from AS 65101, router C creates a single update and processes it through distribute list 20 once, which saves processing time. The update is replicated for each neighbor that is part of the internal peer group.

Thus, the use of peer groups can improve efficiency when processing updates for BGP neighbors that have a common outbound BGP policy.

Adding a new neighbor to router C using a peer group with the same policies of the other IBGP neighbors requires adding only a single neighbor statement to link the new neighbor to the peer group internal. Adding that same neighbor to router C without a peer group requires four neighbor statements.

Using a peer group also makes the configuration easier to read and to change. If you need to add a new policy, such as a route map, to all IBGP neighbors on router C using a peer group, you need only to link the route map to peer group internal. For router C without a peer group, you need to add the new policy to each neighbor.



BGP network Command

```
Router(config-router)#
```

```
network network-number [mask network-mask] [route-map
map-tag]
```

- **This command tells BGP what network to advertise.**
- **The command does not activate the protocol on an interface.**
- **Without a** mask **option, the command advertises classful networks. If a subnet of the classful network exists in a routing table, the classful address is announced.**
- **With the** mask **option, BGP looks for an exact match in the local routing table before announcing the route.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-19

Use the **network** *network-number* command to permit BGP to advertise a network if it is present in the IP routing table.

The network Command Parameters

The table describes the syntax of the BGP **network** command.

Parameter	Description
network-number	Identifies an IP network to be advertised by BGP.
network-mask	(Optional) Identifies the subnet mask to be advertised by BGP.
map-tag	(Optional) Identifier of a configured route map. The route map is examined to filter the networks to be advertised. If not specified, all networks are advertised. If the **route-map** keyword is specified, but no route map tags is listed, no networks will be advertised.

The **network** command determines which networks that the router originates. This concept is different from using the **network** command when you are configuring an IGP. Unlike an IGP, the **network** command does not start BGP on specific interfaces; rather, it indicates to BGP which networks it should originate from this router.

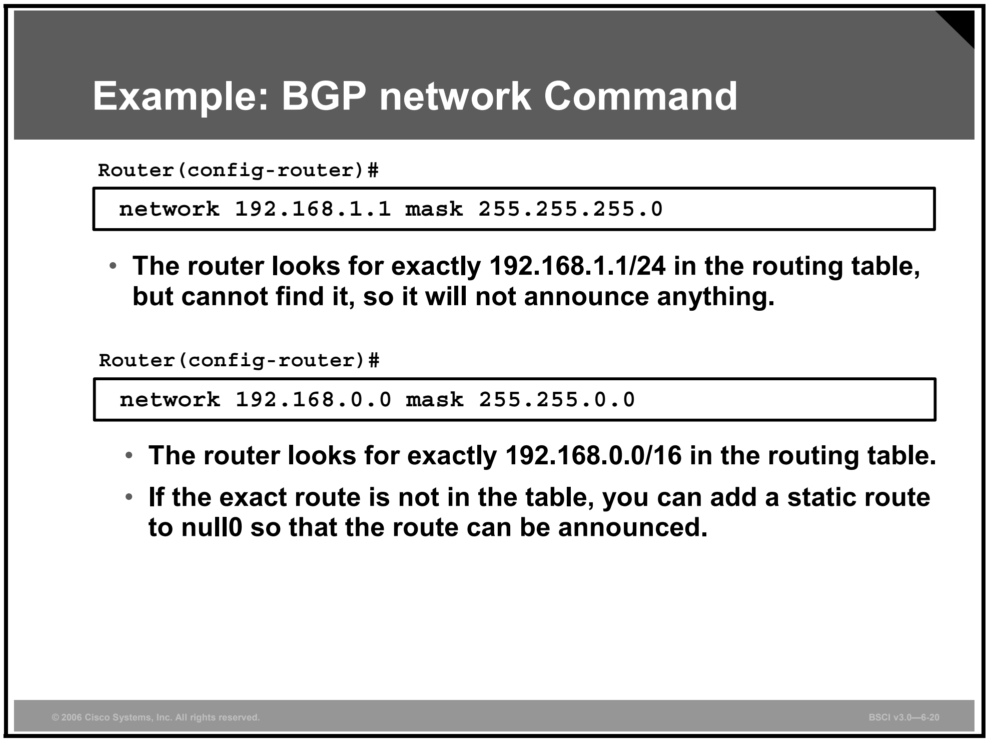
The **mask** parameter indicates that BGP version 4 (BGP4) can handle subnetting and supernetting. The list of **network** commands must include all networks in your AS that you want to advertise, not just those that are locally connected to the router.

Prior to Cisco IOS Software Release 12.0, there was a limit of 200 **network** commands per BGP router. This limit has been removed. The resources of the router, such as the configured NVRAM or RAM, determine the maximum number of **network** commands that you can use.

The **neighbor** command tells BGP where to advertise, and the **network** command tells BGP what to advertise.

The sole purpose of the **network** command is to notify BGP which network to advertise. Without the mask option, this command announces only the classful network number. At least one subnet of the specified major network must be present in the IP routing table to allow BGP to start announcing the classful network as a BGP route.

However, if you specify a *network-mask* option, an exact match to the network (both address and mask) must exist in the routing table. Before BGP announces a route, it checks to see whether it can reach it.



Example: BGP network Command

```
Router(config-router)#
network 192.168.1.1 mask 255.255.255.0
```

- **The router looks for exactly 192.168.1.1/24 in the routing table, but cannot find it, so it will not announce anything.**

```
Router(config-router)#
network 192.168.0.0 mask 255.255.0.0
```

- **The router looks for exactly 192.168.0.0/16 in the routing table.**
- **If the exact route is not in the table, you can add a static route to null0 so that the route can be announced.**

BSCI v3.0—6-20

Example: BGP network Command

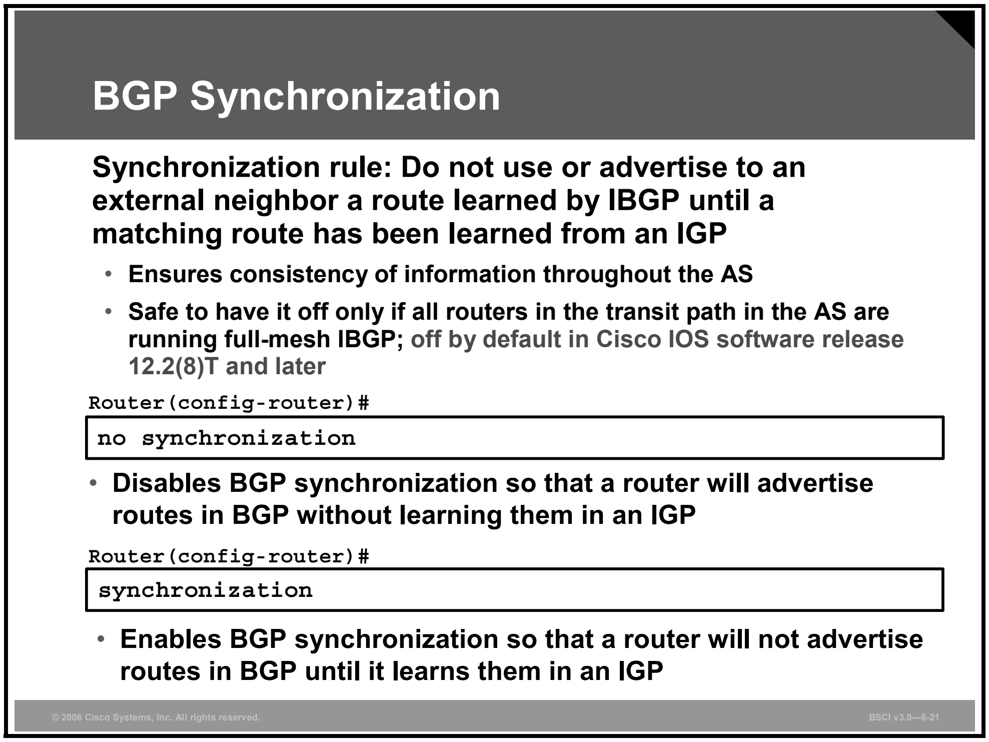
For example, if you misconfigure the command **network 192.168.1.1 mask 255.255.255.0**, BGP looks for exactly 192.168.1.1/24 in the routing table. It may find 192.168.1.0/24 or 192.168.1.1/32; however, it never finds 192.168.1.1/24. Because the routing table does not contain a specific match to the network, BGP does not announce the 198.1.1.1/24 network to any neighbors.

As another example, if you specify **network 192.168.0.0 mask 255.255.0.0** to advertise a classless interdomain routing (CIDR) block, BGP looks for 192.168.0.0/16 in the routing table. It may find 192.168.1.0/24 or 192.168.1.1/32; however, if it never finds 192.168.0.0/16, BGP does not announce the 192.168.0.0/16 network to any neighbors. In this case, you can configure the following static route toward the null interface so BGP can find an exact match in the routing table:

```
ip route 198.1.0.0 255.255.0.0 null0
```

After finding an exact match in the routing table, BGP announces the 192.168.0.0/16 network to any neighbors.

Note The BGP **auto-summary** router configuration command determines how BGP handles redistributed routes. With BGP summarization enabled (with **auto-summary**), all redistributed subnets are summarized to their classful boundaries in the BGP table. When disabled (with **no auto-summary**), all redistributed subnets are present in their original form in the BGP table, so only those subnets would be advertised. In Cisco IOS Software Release 12.2(8)T, the default behavior of the **auto-summary** command was changed to disabled (**no auto-summary**); prior to that the default was enabled (**auto-summary**).



BGP Synchronization

Synchronization rule: Do not use or advertise to an external neighbor a route learned by IBGP until a matching route has been learned from an IGP

- **Ensures consistency of information throughout the AS**
- **Safe to have it off only if all routers in the transit path in the AS are running full-mesh IBGP; off by default in Cisco IOS software release 12.2(8)T and later**

```
Router(config-router)#
no synchronization
```

- **Disables BGP synchronization so that a router will advertise routes in BGP without learning them in an IGP**

```
Router(config-router)#
synchronization
```

- **Enables BGP synchronization so that a router will not advertise routes in BGP until it learns them in an IGP**

BSCI v3.0—6-21

The BGP synchronization rule states that a BGP router should not use, or advertise to an external neighbor, a route that is learned from IBGP unless that route is local or the router learns it from the IGP. In other words, BGP and the IGP must be synchronized before the networks learned from an IBGP neighbor can be used.

If an AS passes traffic to another AS, BGP should not advertise a route before all routers in the AS have learned about the route via the IGP. A router learning a route via IBGP waits until the IGP has propagated the route within the AS and then advertises it to external peers. This rule ensures that all routers in the AS are synchronized and are able to route traffic that the AS advertises to other autonomous systems.

This approach ensures consistency of routing information (avoids “black holes”) within the AS.

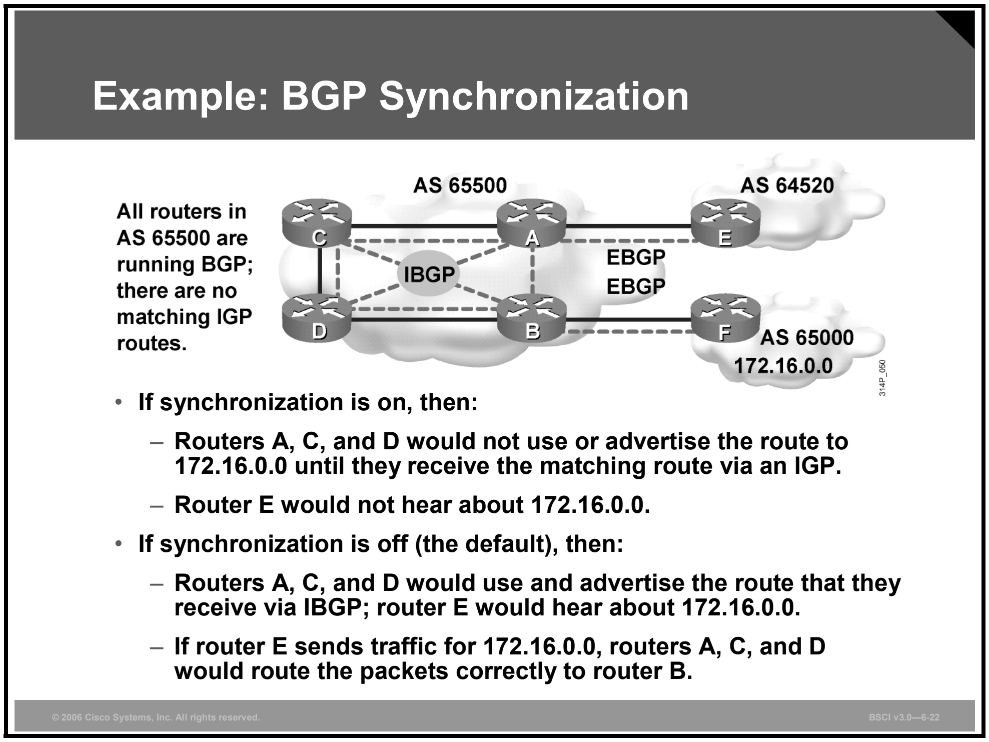
BGP synchronization is disabled by default in Cisco IOS Software Release 12.2(8)T and later; it was on by default in earlier Cisco IOS software releases. With the default of synchronization disabled, BGP can use and advertise to an external BGP neighbor routes learned from an IBGP neighbor that are not present in the local routing table

BGP synchronization is unnecessary in some situations. It is safe to have BGP synchronization off only if all routers in the transit path in the AS are running full-mesh IBGP.

Having synchronization disabled allows the routers to carry fewer routes in IGP and allows BGP to converge more quickly.

Use synchronization if there are routers in the BGP transit path in the AS that are not running BGP (therefore, the routers do not have full-mesh IBGP within the AS).

Note In the past, the best practice was to redistribute BGP into the IGP running in an AS, so that IBGP was not needed in every router in the transit path. In this case, synchronization was needed to make sure that packets did not get lost, so synchronization was on by default. As the Internet grew, the number of routes in the BGP table became too much for the IGPs to handle, so the best practice changed to not redistributing BGP into the IGP, but instead using IBGP on all routers in the transit path. In this case, synchronization is not needed; it is now off by default.



Example: BGP Synchronization

All routers in AS 65500 are running BGP; there are no matching IGP routes.

- If synchronization is on, then:
  - Routers A, C, and D would not use or advertise the route to 172.16.0.0 until they receive the matching route via an IGP.
  - Router E would not hear about 172.16.0.0.
- If synchronization is off (the default), then:
  - Routers A, C, and D would use and advertise the route that they receive via IBGP; router E would hear about 172.16.0.0.
  - If router E sends traffic for 172.16.0.0, routers A, C, and D would route the packets correctly to router B.

BSCI v3.0—6-22

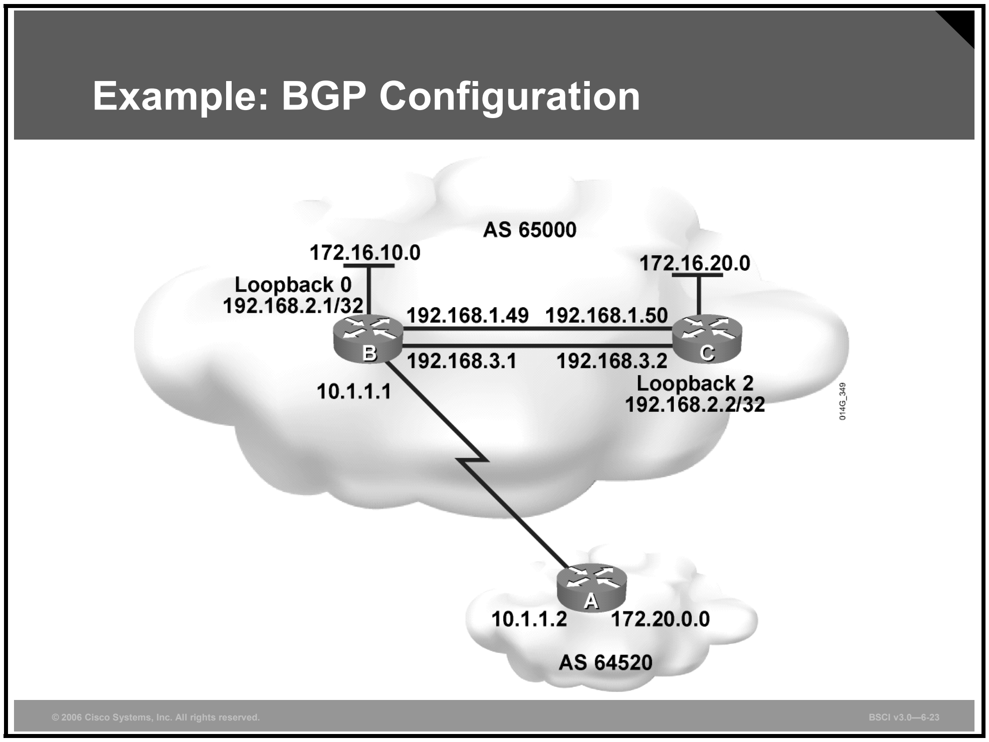
Example: BGP Synchronization

In this figure, routers A, B, C, and D are all running IBGP and an IGP with each other. There are no matching IGP routes for the BGP routes (routers A and B are not redistributing the BGP routes into the IGP). Routers A, B, C, and D have IGP routes to the internal networks of AS 65500 but do not have routes to external networks such as 172.16.0.0.

Router B advertises the route to 172.16.0.0 to the other routers in AS 65500 using IBGP. If synchronization is on, routers A, C, and D do not use the route to 172.16.0.0, nor does router A advertise that route to router E in AS 64520. Router B uses the route to 172.16.0.0 and installs it in its routing table. If router E receives traffic that is destined for network 172.16.0.0, it does not have a route for that network and cannot forward the traffic.

If synchronization is off (the default) in AS 65500, routers A, C, and D can use the route to 172.16.0.0 and install the route in their routing tables even if there are no matching IGP routes for the BGP routes (assuming that routers A, C, and D can reach the next-hop address for 172.16.0.0). Router A advertises the route to router E. Router E then has a route to 172.16.0.0 and may send traffic that is destined for that network. Router E sends the packets to router A, and router A forwards them to router C. Router C learns a route to 172.16.0.0 via IBGP; therefore, router C forwards the packets to router D. Router D forwards the packets to router B. Router B forwards the packets to router F for network 172.16.0.0.

In modern autonomous systems, because the size of the Internet routing table is large, redistributing from BGP into an IGP is not scalable; therefore, most modern autonomous systems run full-mesh IBGP and do not require synchronization. Advanced BGP configuration methods, for example, using route reflectors and confederations, reduce the full-mesh requirements.

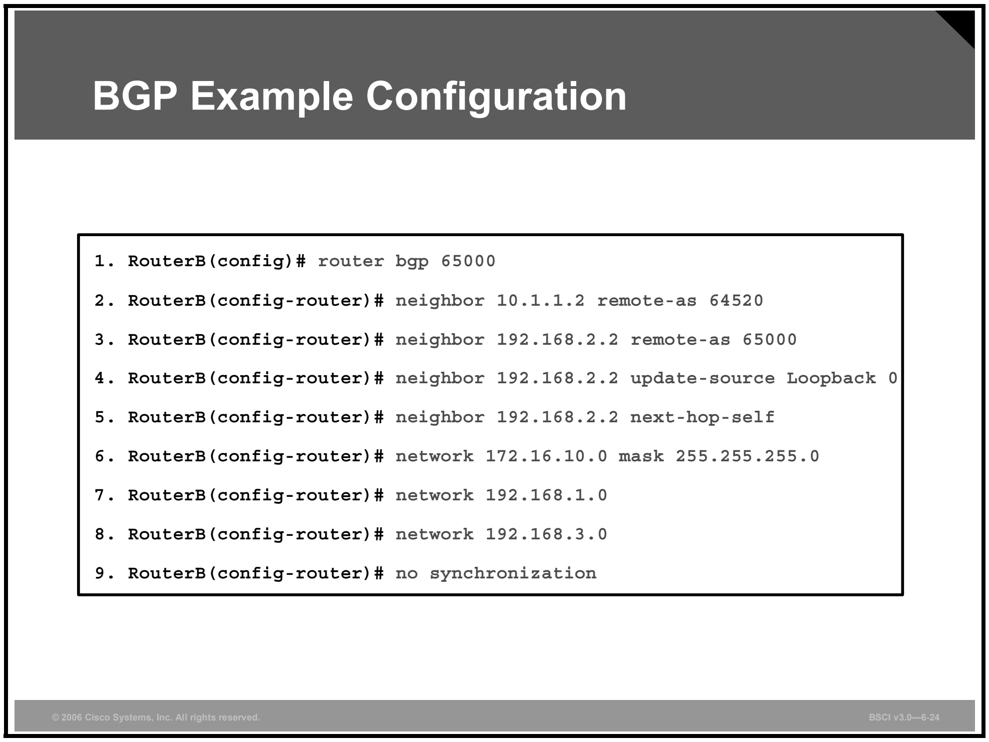


Example: BGP Configuration

BSCI v3.0—6-23

Example: BGP Configuration

This figure shows another BGP example. The configuration for router B follows.



BGP Example Configuration

```
1. RouterB(config)# router bgp 65000
2. RouterB(config-router)# neighbor 10.1.1.2 remote-as 64520
3. RouterB(config-router)# neighbor 192.168.2.2 remote-as 65000
4. RouterB(config-router)# neighbor 192.168.2.2 update-source Loopback 0
5. RouterB(config-router)# neighbor 192.168.2.2 next-hop-self
6. RouterB(config-router)# network 172.16.10.0 mask 255.255.255.0
7. RouterB(config-router)# network 192.168.1.0
8. RouterB(config-router)# network 192.168.3.0
9. RouterB(config-router)# no synchronization
```

BSCI v3.0—6-24

Example: BGP Configuration for Router B

This figure shows the configuration for router B. The first two commands under the **router bgp 65000** command establish that router B has the following two BGP neighbors:

- Router A in AS 64520
- Router C in AS 65000

From the perspective of router B, router A is an EBGP neighbor and router C is an IBGP neighbor.

The neighbor statement on router B for router A is pointing at the directly connected IP address to reach the EBGP neighbor, router A. However, the neighbor statement on router B points to the loopback interface of router C because router B has multiple paths to reach router C.

If router B pointed at the 192.168.3.2 IP address of router C and that interface went down, router B would be unable to reestablish the BGP session until the link came back up. By pointing to the loopback interface of router C instead, the link stays established as long as any path to router C is available. Router C should also point to the loopback address of router B in its configuration.

Line 4 notifies router B to always use its Loopback 0 address, 192.168.2.1, as the source IP address when sending an update to router C, 192.168.2.2.

In line 5, router B changes the next-hop address for networks that are reachable through it. The default next-hop setting for networks from AS 64520 is IP address 10.1.1.2. With this **next-hop-self** command, router B sets the next-hop address to the source IP address of the routing update, which is the router B Loopback 0 interface, as set by the **update-source** command.

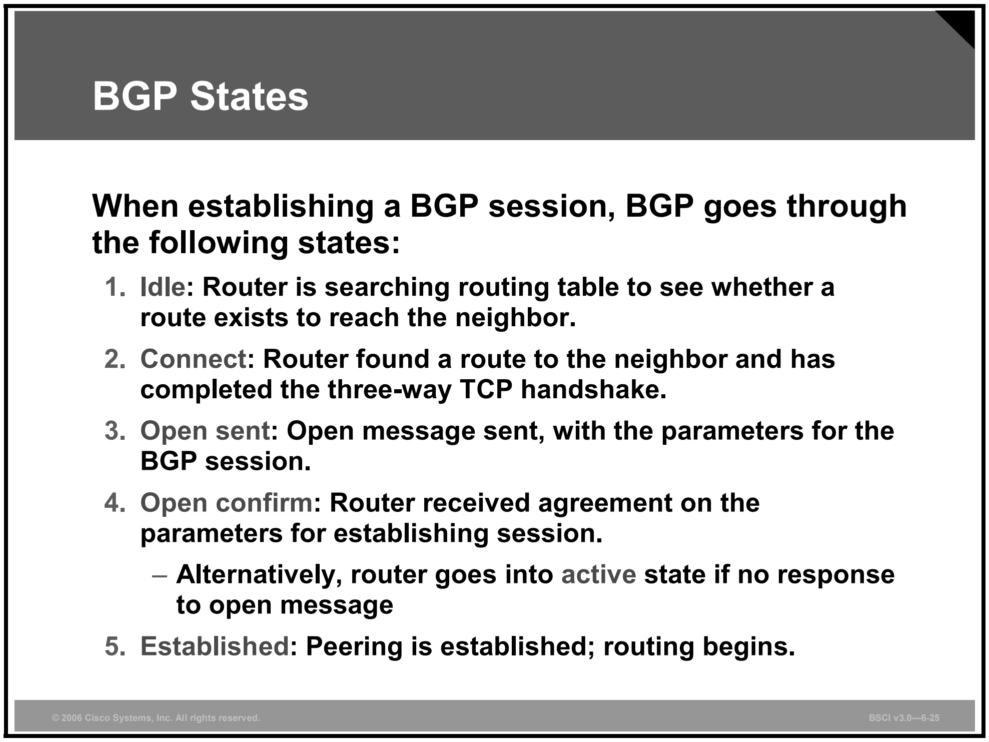
Lines 6 and 7 notify BGP about which networks to advertise. Line 6 contains a subnet of a class B address using the **mask** option. Lines 7 and 8 have two network statements for the two class C networks that connect router B and router C. The default mask is 255.255.255.0, so you do not need to include it in the command.

In line 9, synchronization is disabled (this command is no longer needed if the router is running Cisco IOS Software Release 12.2(8)T or later because it is off by default). If router A is advertising 172.20.0.0 in BGP, router B receives that route and advertises it to router C. Because synchronization is off, router C can use this route.

If router C had EBGP neighbors of its own and router B wanted to use router C as the path to those networks, synchronization on router B would also need to be off. In this network synchronization can be off because all the routers within the AS are running IBGP.

Identifying BGP Neighbor States

This topic describes BGP neighbor states.



BGP States

When establishing a BGP session, BGP goes through the following states:

1. Idle: **Router is searching routing table to see whether a route exists to reach the neighbor.**
2. Connect: **Router found a route to the neighbor and has completed the three-way TCP handshake.**
3. Open sent: **Open message sent, with the parameters for the BGP session.**
4. Open confirm: **Router received agreement on the parameters for establishing session.**
   - **Alternatively, router goes into** active **state if no response to open message**
5. Established: **Peering is established; routing begins.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-25

After the TCP handshake is complete, the BGP application tries to set up a session with the neighbor. A number of steps must occur for the session to establish itself.

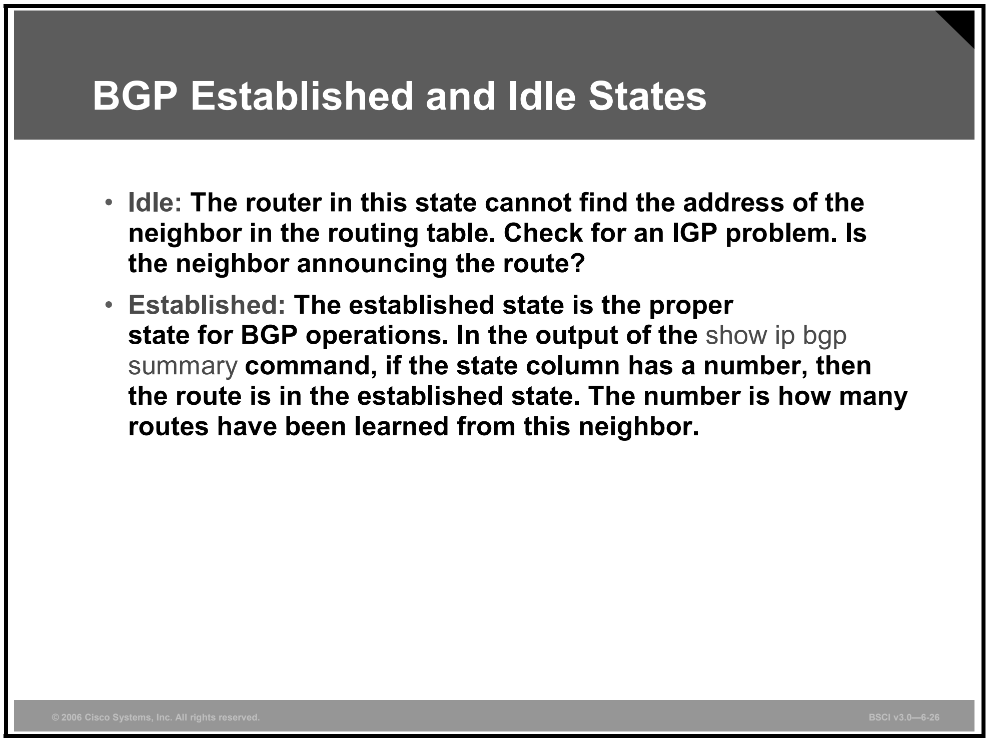
After you have entered the **neighbor** command in BGP, BGP takes the IP address that is listed and checks the local routing table for a route to this address. At this point, BGP is in the idle state. If BGP does not find a route to the IP address, it stays in the idle state. If it finds a route, it goes to the connect state when the TCP handshaking synchronize acknowledge (SYN ACK) packet returns.

After the TCP connection has finished, BGP creates a BGP open packet and sends it to the neighbor. Once BGP dispatches this open packet, the BGP peering session changes to the open sent state. If there is no response for 5 seconds, the state changes to the active state.

If a response does come back in a timely manner, BGP goes to the open confirm state and starts scanning (evaluating) the routing table for the paths to send to the neighbor. When those paths have been found, BGP then goes to the established state and begins routing between the neighbors.

Note	The states that two BGP routers are going through to establish a session can be observed using **debug** commands. In Cisco IOS Software Release 12.4, the **debug ip bgp ipv4 unicast** command can be used to see this process. In earlier Cisco IOS releases, the **debug ip bgp events** command gave similar output.

Note	Debugging uses up router resources and should be turned on only when necessary.



BGP Established and Idle States

- Idle: **The router in this state cannot find the address of the neighbor in the routing table. Check for an IGP problem. Is the neighbor announcing the route?**
- Established: **The established state is the proper state for BGP operations. In the output of the** show ip bgp summary **command, if the state column has a number, then the route is in the established state. The number is how many routes have been learned from this neighbor.**

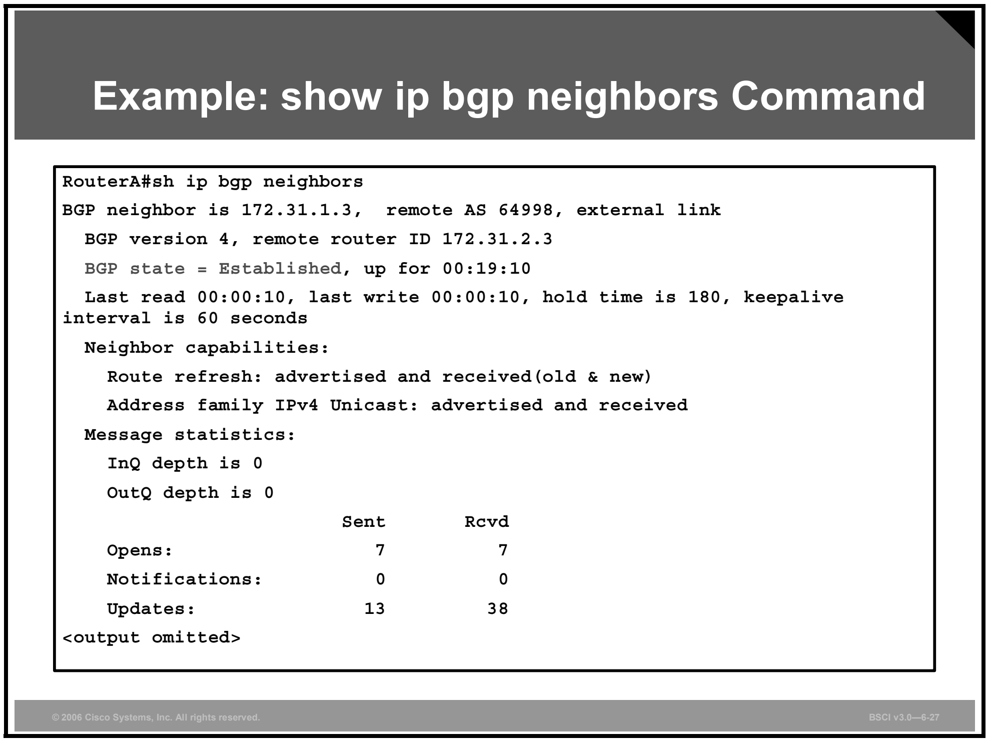
The idle state is an indication that the router does not know how to reach the IP address that is listed in the neighbor statement. The router is idle because of one of the following scenarios:

- It is waiting for a static route to that IP address or network to be configured.
- It is waiting for the local routing protocol (IGP) to learn about this network through an advertisement from another router.

The most common reason for a router to enter the idle state is that the neighbor is not announcing the IP address or network that the neighbor statement of the router is pointing to. Check these two conditions first to correct this problem:

- Ensure that the neighbor announces the route in its local routing protocol (IGP).
- Verify that you have not entered an incorrect IP address in the neighbor statement.

The established state is the desired state for the neighbor relationship. This state means that both routers have agreed to exchange BGP updates with one another and routing has begun.

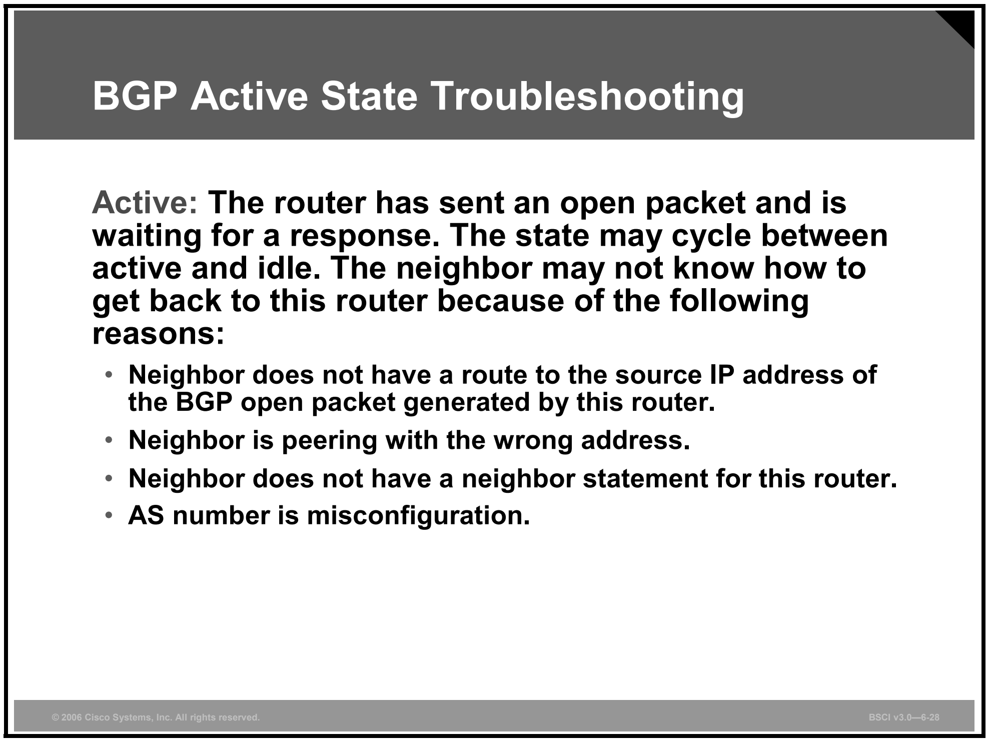


Example: show ip bgp neighbors Command

```
RouterA#sh ip bgp neighbors
BGP neighbor is 172.31.1.3,  remote AS 64998, external link
  BGP version 4, remote router ID 172.31.2.3
  BGP state = Established, up for 00:19:10
  Last read 00:00:10, last write 00:00:10, hold time is 180, keepalive
interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
                        Sent       Rcvd
    Opens:                 7          7
    Notifications:         0          0
    Updates:              13         38
<output omitted>
```

Example: show ip bgp neighbors Command

Use the **show ip bgp neighbors** command to display information about the BGP connections to neighbors. In the figure, the BGP state is established, which means that the neighbors have established a TCP connection and the two peers have agreed to use BGP to communicate.



BGP Active State Troubleshooting

Active: The router has sent an open packet and is waiting for a response. The state may cycle between active and idle. The neighbor may not know how to get back to this router because of the following reasons:

- **Neighbor does not have a route to the source IP address of the BGP open packet generated by this router.**
- **Neighbor is peering with the wrong address.**
- **Neighbor does not have a neighbor statement for this router.**
- **AS number is misconfiguration.**

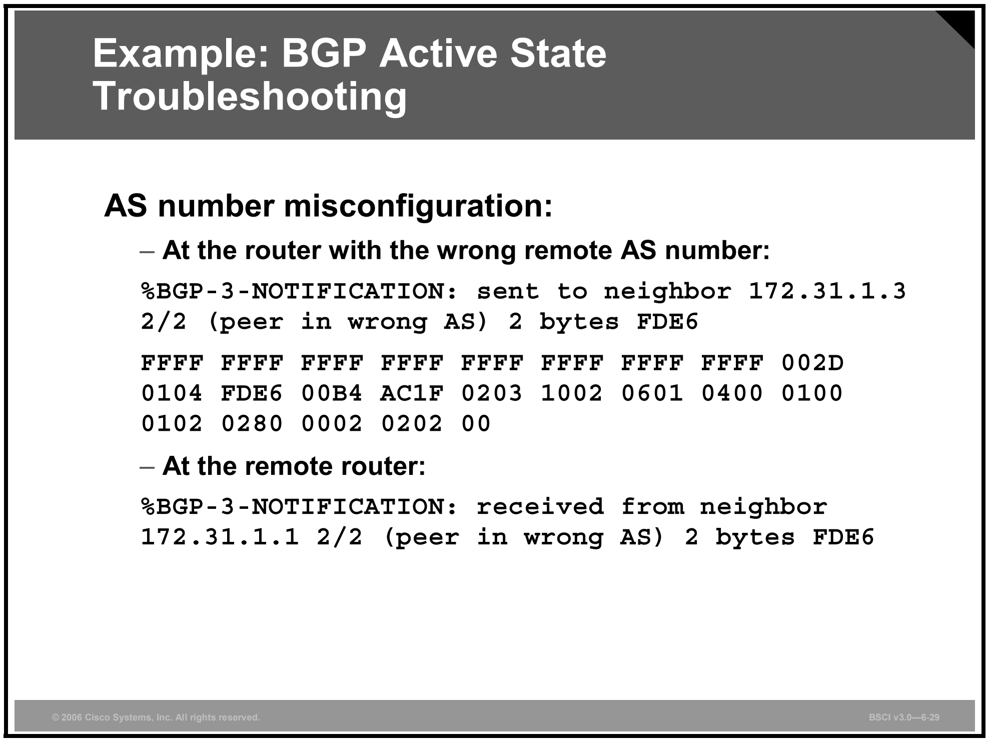
BSCI v3.0—6-28

If the router is in the active state, it means that it has found the IP address in the neighbor statement and has created and sent out a BGP open packet. However, the router has not received a response (open confirm packet) back.

One common problem in this case is that the neighbor may not have a return route to the source IP address. Ensure that the source IP address or network of the packets has been announced to the local routing protocol (IGP).

Another common problem associated with the active state occurs when a BGP router attempts to peer with another BGP router that does not have a neighbor statement peering back at the first router, or when the other router is peering with the wrong IP address on the first router. Check to ensure that the other router has a neighbor statement peering at the correct address of the router that is in the active state.

If the state toggles between the idle state and the active state, one of the most common problems is AS number misconfiguration.



Example: BGP Active State Troubleshooting

AS number misconfiguration:

– **At the router with the wrong remote AS number:**

```
%BGP-3-NOTIFICATION: sent to neighbor 172.31.1.3
2/2 (peer in wrong AS) 2 bytes FDE6

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D
0104 FDE6 00B4 AC1F 0203 1002 0601 0400 0100
0102 0280 0002 0202 00
```

– **At the remote router:**

```
%BGP-3-NOTIFICATION: received from neighbor
172.31.1.1 2/2 (peer in wrong AS) 2 bytes FDE6
```

BSCI v3.0—6-29

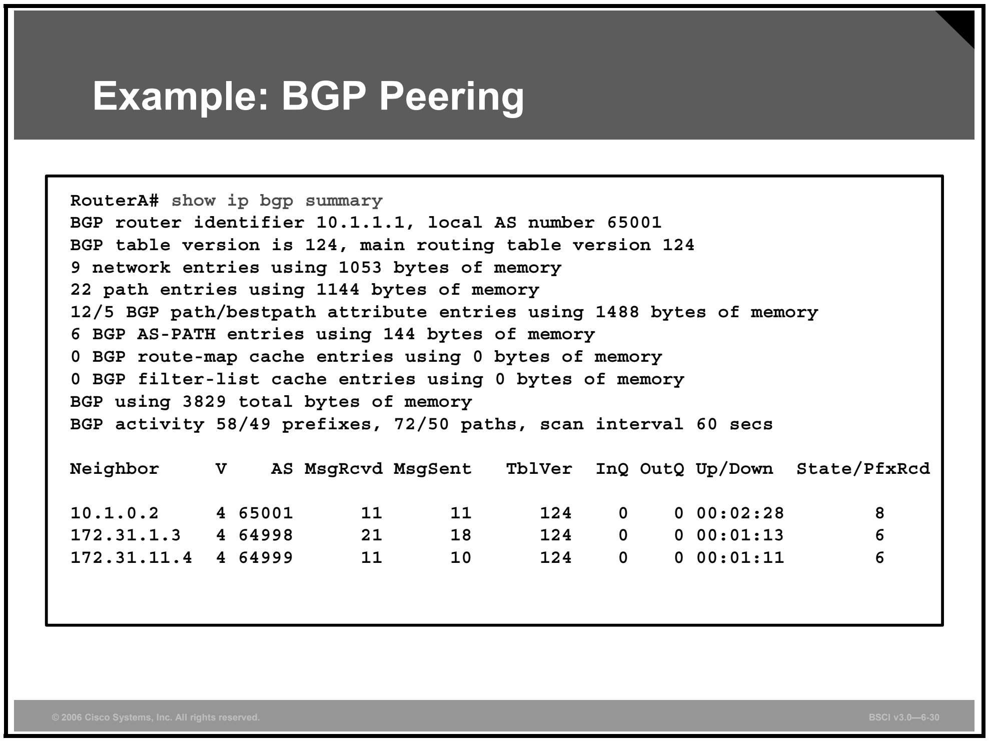
Example: BGP Active State Troubleshooting

If you have misconfigured the AS number, you will see the following console message at the router with the wrong remote AS number configured in the neighbor statement:

```
%BGP-3-NOTIFICATION: sent to neighbor 172.31.1.3 2/2 (peer in
wrong AS) 2 bytes FDE6

FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 002D 0104 FDE6 00B4
AC1F 0203 1002 0601 0400 0100 0102 0280 0002 0202 00
```

At the remote router, you will see the following message:

```
%BGP-3-NOTIFICATION: received from neighbor 172.31.1.1 2/2
(peer in wrong AS) 2 bytes FDE6
```



Example: BGP Peering

```
RouterA# show ip bgp summary
BGP router identifier 10.1.1.1, local AS number 65001
BGP table version is 124, main routing table version 124
9 network entries using 1053 bytes of memory
22 path entries using 1144 bytes of memory
12/5 BGP path/bestpath attribute entries using 1488 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3829 total bytes of memory
BGP activity 58/49 prefixes, 72/50 paths, scan interval 60 secs

Neighbor     V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

10.1.0.2     4 65001      11      11      124    0    0 00:02:28        8
172.31.1.3   4 64998      21      18      124    0    0 00:01:13        6
172.31.11.4  4 64999      11      10      124    0    0 00:01:11        6
```

BSCI v3.0—6-30

Example: BGP Peering

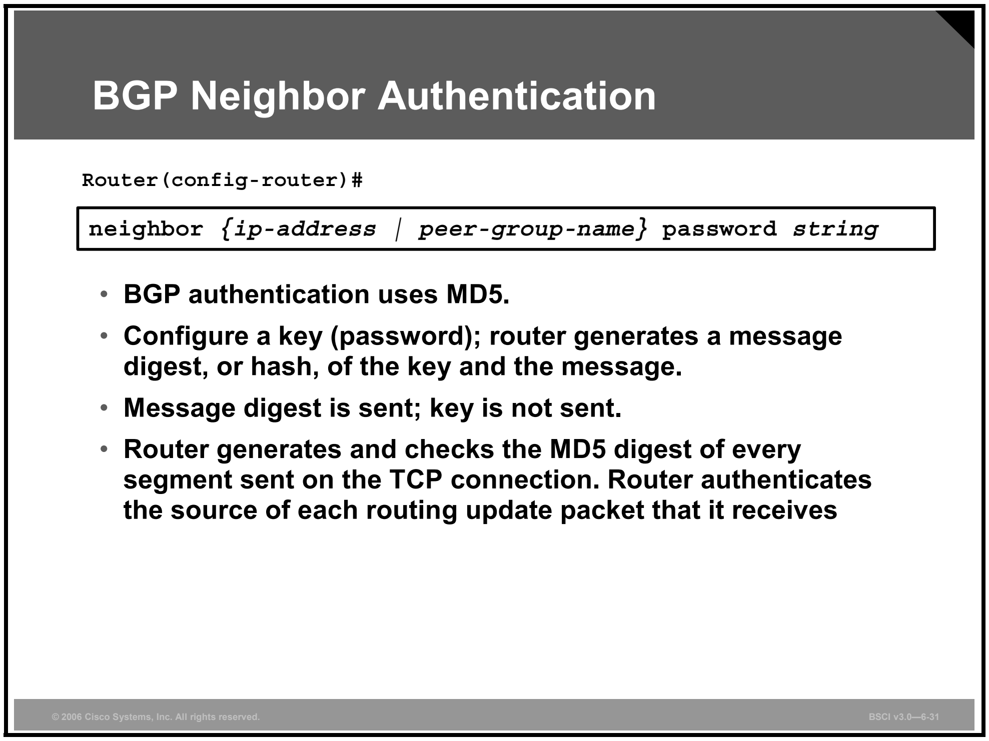
The **show ip bgp summary** command is one way to verify the neighbor relationship. The figure presents the output from this command. Some of the details of this command output are as follows:

- **BGP router ID:** IP address that all other BGP speakers recognize as representing this router.
- **BGP table version:** Increases in increments when the BGP table changes.
- **Main routing table version:** Last version of the BGP database that was injected into the main routing table.
- **Neighbor:** The IP address that is used in the neighbor statement with which this router has a relationship.
- **Version (V):** The version of BGP that this router is running with the listed neighbor.
- **AS:** The AS number of the listed neighbor.
- **Messages received (MsgRcvd):** The number of BGP messages that have been received from this neighbor.
- **Messages sent (MsgSent):** The number of BGP messages sent to this neighbor.
- **Table version (TblVer):** BGP table version.
- **In queue (InQ):** The number of messages waiting to be processed from this neighbor.
- **Out queue (OutQ):** The number of messages queued and waiting to be sent to this neighbor. TCP flow control prevents this router from overwhelming a neighbor with a large update.

- **Up/Down:** The length of time that this neighbor has been in the current BGP state (established, active, or idle).
- **State [established, active, idle, open sent, open confirm, or idle (admin)]:** The BGP state. You can set a neighbor to administratively shut down (admin state) by using the **neighbor shutdown** router configuration command.
- **Prefix received (PfxRcd):** When the session is in the established state, this value represents the number of BGP network entries received from the listed neighbor.

Authenticating in BGP

This topic describes how to configure Message Digest 5 (MD5) authentication on the BGP TCP connection between two routers.



BGP Neighbor Authentication

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name} password string
```

- **BGP authentication uses MD5.**
- **Configure a key (password); router generates a message digest, or hash, of the key and the message.**
- **Message digest is sent; key is not sent.**
- **Router generates and checks the MD5 digest of every segment sent on the TCP connection. Router authenticates the source of each routing update packet that it receives**

BGP neighbor authentication can be configured on a router so that the router authenticates the source of each routing update packet that it receives. This is accomplished by the exchange of an authenticating key (sometimes referred to as a password) that is known to both the sending and the receiving router.

BGP supports MD5 neighbor authentication. MD5 sends a message digest (also called a hash) that is created using the key and a message. The message digest is then sent instead of the key. The key itself is not sent, to prevent it from being read by an eavesdropper on the line while it is being transmitted.

To enable MD5 authentication on a TCP connection between two BGP peers, use the **neighbor** {*ip-address* | *peer-group-name*} **password** *string* router configuration command.

The neighbor password Command Parameters

The table describes the syntax of the **neighbor password** command.

Parameter	Description
ip-address	IP address of the BGP-speaking neighbor.
peer-group-name	Name of a BGP peer group.
string	Case-sensitive password of up to 25 characters. The first character cannot be a number. The string can contain any alphanumeric characters, including spaces. You cannot specify a password in the format *number-space-anything*. The space after the number can cause authentication to fail.

You can configure MD5 authentication between two BGP peers, meaning that each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both BGP peers; otherwise, the connection between them will not be made. Configuring MD5 authentication causes Cisco IOS software to generate and check the MD5 digest of every segment sent on the TCP connection.

Caution If the authentication string is configured incorrectly, the BGP peering session will not be established. It is recommended that you enter the authentication string carefully and verify that the peering session is established after authentication is configured.

If you specify a BGP peer group by using the *peer-group-name* argument, all the members of the peer group will inherit the characteristic configured with this command.

If a router has a password configured for a neighbor, but the neighbor router does not, a message such as the following will appear on the console when the routers attempt to send BGP messages between themselves:

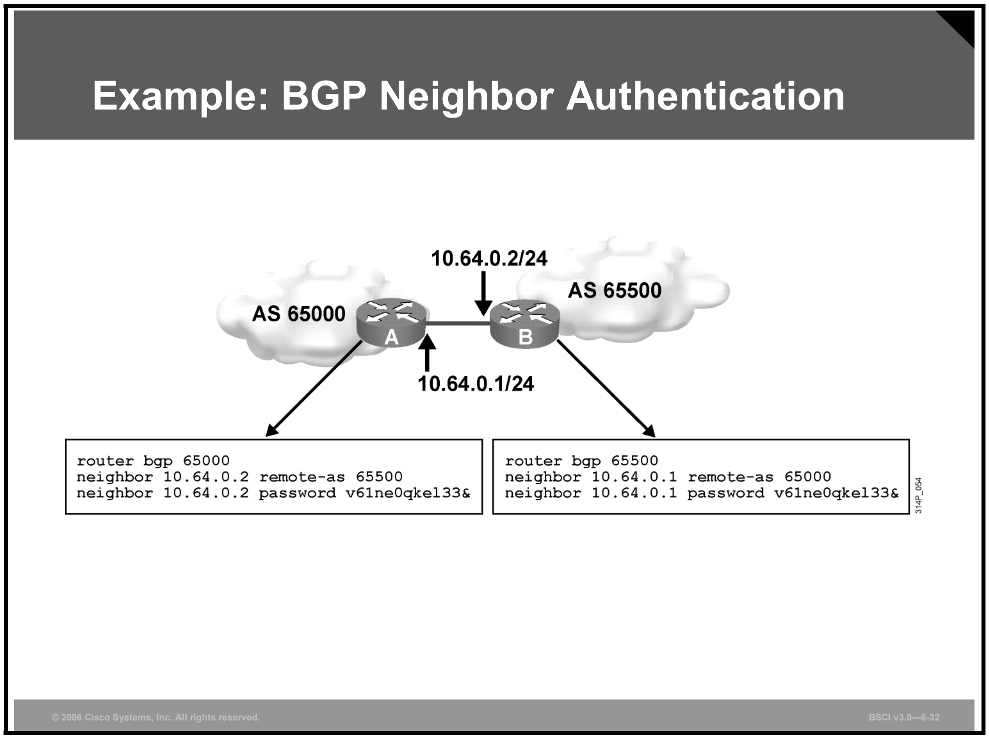
```
%TCP-6-BADAUTH: No MD5 digest from 10.1.0.2(179) to
10.1.0.1(20236)
```

Similarly, if the two routers have different passwords configured, a message such as the following will appear on the screen:

```
%TCP-6-BADAUTH: Invalid MD5 digest from 10.1.0.1(12293) to
10.1.0.2(179)
```

If you configure or change the password or key used for MD5 authentication between two BGP peers, the local router will not tear down the existing session after you configure the password. The local router will attempt to maintain the peering session using the new password until the BGP hold-down timer expires. The default time period is 180 seconds. If the password is not entered or changed on the remote router before the hold-down timer expires, the session times out.

Note Configuring a new timer value for the holddown timer will only take effect after the session has been reset. It is not possible to change the configuration of the hold-down timer to avoid resetting the BGP session.



Example: BGP Neighbor Authentication

```
router bgp 65000
neighbor 10.64.0.2 remote-as 65500
neighbor 10.64.0.2 password v61ne0qkel33&
```

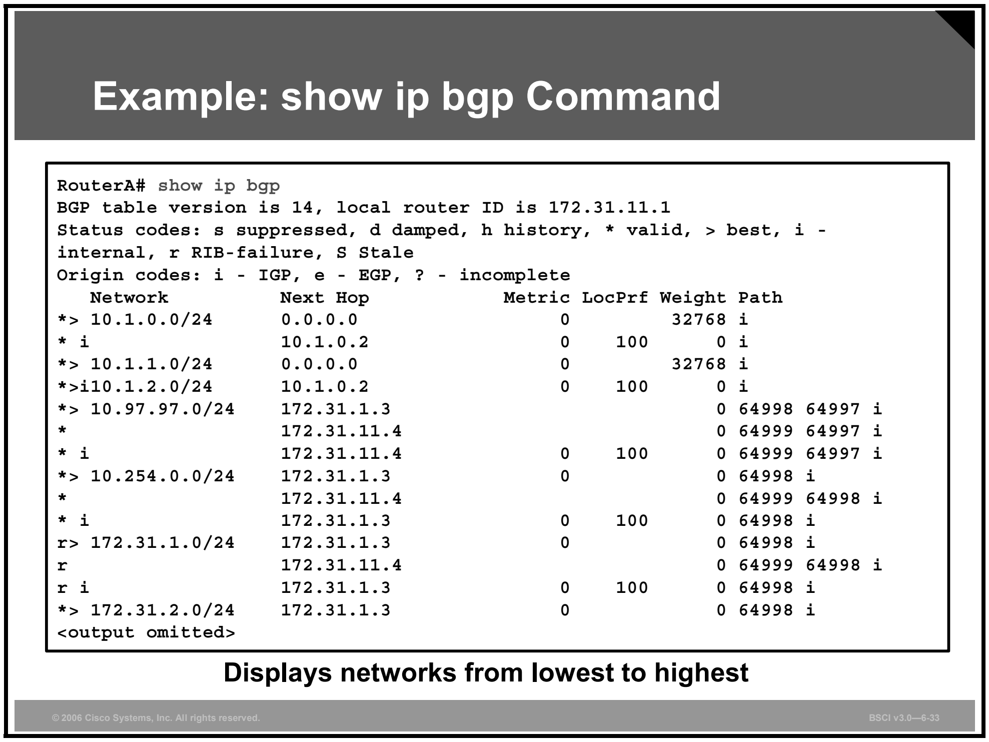
```
router bgp 65500
neighbor 10.64.0.1 remote-as 65000
neighbor 10.64.0.1 password v61ne0qkel33&
```

Example: BGP Neighbor Authentication

The example in the figure configures MD5 authentication for the BGP peering session between routers A and B. The same password must be configured on the remote peer before the hold-down timer expires.

Troubleshooting BGP

This topic describes how to troubleshoot BGP.



Example: show ip bgp Command

```
RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/24      0.0.0.0                  0         32768 i
* i                 10.1.0.2                 0    100      0 i
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*>i10.1.2.0/24      10.1.0.2                 0    100      0 i
*> 10.97.97.0/24    172.31.1.3                             0 64998 64997 i
*                   172.31.11.4                            0 64999 64997 i
* i                 172.31.11.4              0    100      0 64999 64997 i
*> 10.254.0.0/24    172.31.1.3               0             0 64998 i
*                   172.31.11.4                            0 64999 64998 i
* i                 172.31.1.3               0    100      0 64998 i
r> 172.31.1.0/24    172.31.1.3               0             0 64998 i
r                   172.31.11.4                            0 64999 64998 i
r i                 172.31.1.3               0    100      0 64998 i
*> 172.31.2.0/24    172.31.1.3               0             0 64998 i
<output omitted>
```

Displays networks from lowest to highest

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-33

Example: show ip bgp Command Output

Use the **show ip bgp** command to display the BGP topology database (BGP table).

The figure is a partial sample output of the **show ip bgp** command. The status codes are shown at the beginning of each line of output, and the origin codes are shown at the end of each line. In this output, there is an asterisk (*) in most of the entries in the first column. This means that the next-hop address (in the fifth column) is valid. The next-hop address is not always the router that is directly connected to this router. Other options for the first column are as follows:

- An “s” indicates that the specified routes are suppressed (usually because routes have been summarized and only the summary route is being sent)
- A “d,” for dampening, indicates that the route is being dampened (penalized) for going up and down too often. Although the route might be up right now, it is not advertised until the penalty has expired.
- An “h,” for history, indicates that the route is unavailable and is probably down; historic information about the route exists, but a best route does not exist.
- An “r,” for routing information base (RIB) failure, indicates that the route was not installed in the RIB. The reason that the route is not installed can be displayed using the **show ip bgp rib-failure** command, as shown in the next figure.
- An “S,” for stale, indicates that the route is stale (this symbol is used in the nonstop forwarding-aware router).

The second column shows “>” when BGP has selected the path as the best path to a network.

The third column is either blank or shows “i.” If it is blank, BGP learned that route from an external peer. An “i” indicates that an IBGP neighbor advertised this path to the router.

The fourth column lists the networks that the router learned.

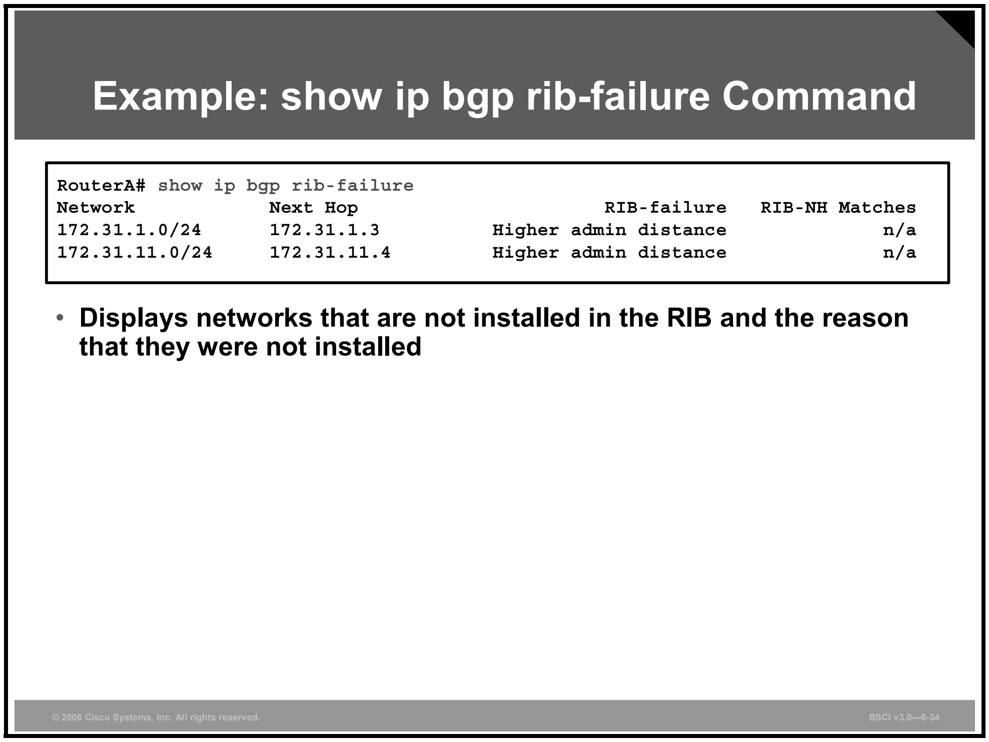
The Next Hop column lists all the next-hop addresses for each route. This column may contain the entry 0.0.0.0, which signifies that this router is the originator of the route.

The three columns to the left of the Path column list three BGP path attributes that are associated with the path: metric (multi-exit discriminator [MED]), local preference, and weight.

The column with the Path header may contain a sequence of autonomous systems in the path. From left to right, the first AS listed is the adjacent AS that this network was learned from. The last number (the rightmost AS number) is the originating AS of this network. The AS numbers between these two represent the exact path that a packet takes back to the originating AS. If the path column is blank, the route is from the current AS.

The last column signifies how this route was entered into BGP on the original router. If the last column has an “i” in it, the originating router probably used a network statement to introduce this network into BGP.

If the character is an “e,” the originating router learned this network from Exterior Gateway Protocol (EGP), which is the historical predecessor to BGP. A question mark (?) signifies that BGP cannot absolutely verify the availability of this network because it is redistributed from an IGP into BGP.



Example: show ip bgp rib-failure Command

```
RouterA# show ip bgp rib-failure
Network            Next Hop                      RIB-failure   RIB-NH Matches
172.31.1.0/24      172.31.1.3          Higher admin distance              n/a
172.31.11.0/24     172.31.11.4         Higher admin distance              n/a
```

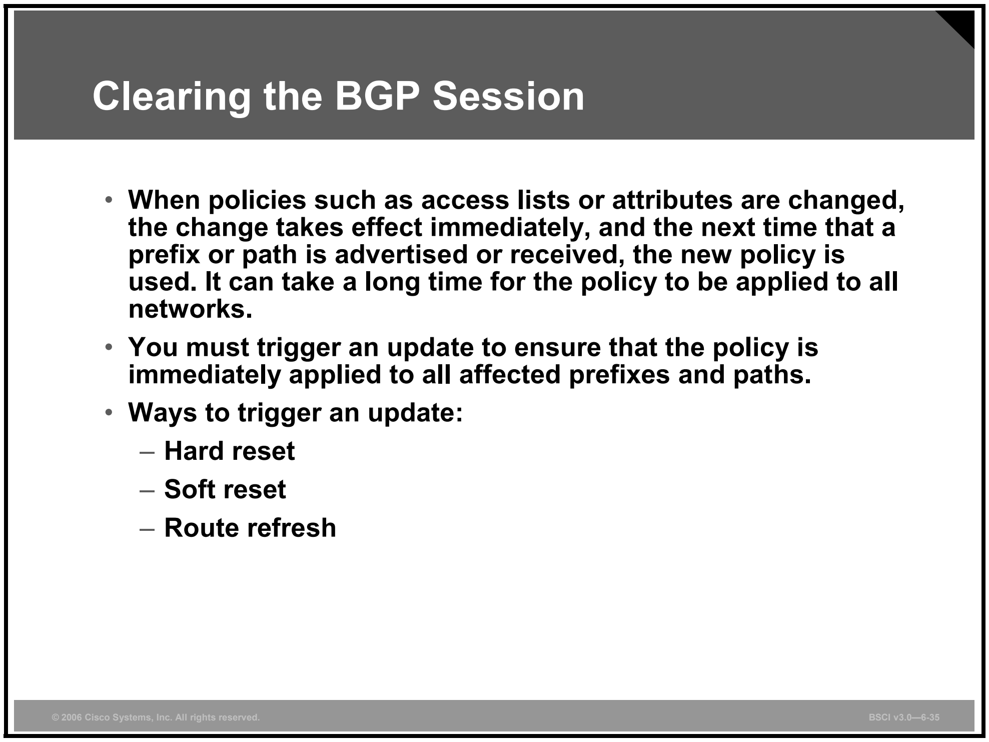
- **Displays networks that are not installed in the RIB and the reason that they were not installed**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-34

Example: show ip bgp rib-failure Command Output

Use the **show ip bgp rib-failure** command to display BGP routes that were not installed in the RIB and the reason that they were not installed.

The example in the figure shows that the displayed routes were not installed because a route or routes with a better administrative distance already exist in the RIB.



Clearing the BGP Session

- **When policies such as access lists or attributes are changed, the change takes effect immediately, and the next time that a prefix or path is advertised or received, the new policy is used. It can take a long time for the policy to be applied to all networks.**
- **You must trigger an update to ensure that the policy is immediately applied to all affected prefixes and paths.**
- **Ways to trigger an update:**
  - **Hard reset**
  - **Soft reset**
  - **Route refresh**

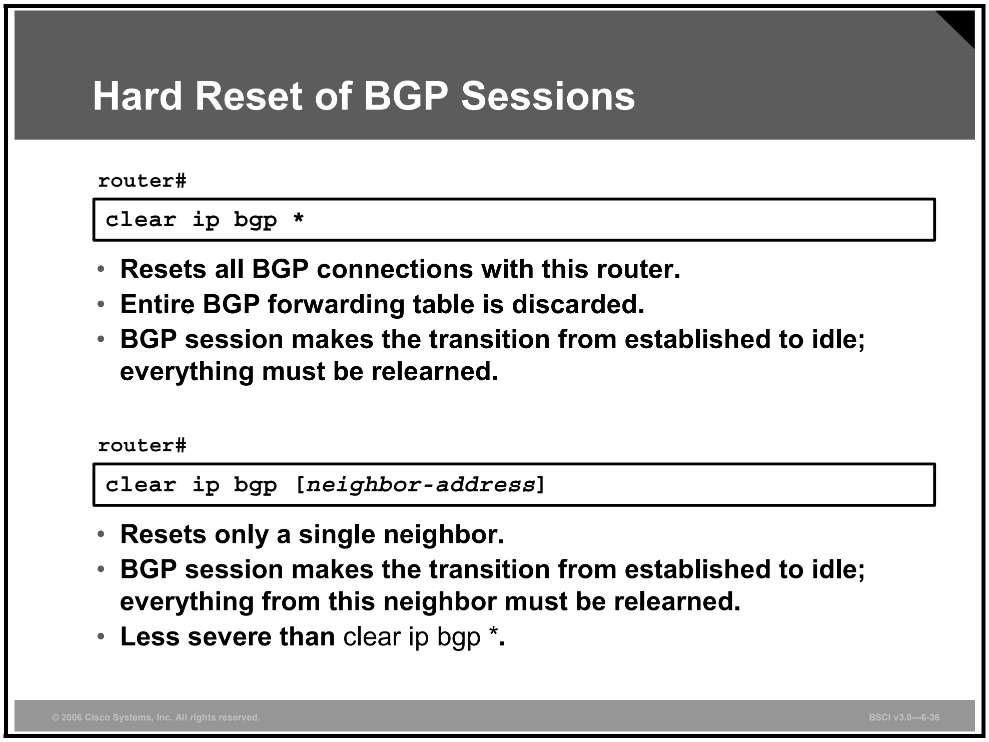
BSCI v3.0—6-35

BGP can potentially handle huge volumes of routing information. When a policy configuration change occurs, the router cannot go through the huge table of BGP information and recalculate which entry is no longer valid in the local table. Nor can the router determine which route or routes, already advertised, should be withdrawn from a neighbor.

There is an obvious risk that the first configuration change will be immediately followed by a second, which would cause the whole process to start all over again. To avoid such a problem, Cisco IOS software applies changes only to those updates that are received or transmitted after the BGP policy configuration change has been performed. The new policy, enforced by the new filters, is applied only on routes that are received or sent after the change.

A network administrator who would like the policy change to be applied on all routes must trigger an update to force the router to let all routes pass through the new filter. If the filter is applied on outgoing information, the router has to resend the BGP table through the new filter. If the filter is applied on incoming information, the router needs its neighbor to resend its BGP table so that it passes through the new filters.

There are three ways to trigger an update: with a hard reset, soft reset, or route refresh.



Hard Reset of BGP Sessions

```
router#
clear ip bgp *
```

- **Resets all BGP connections with this router.**
- **Entire BGP forwarding table is discarded.**
- **BGP session makes the transition from established to idle; everything must be relearned.**

```
router#
clear ip bgp [neighbor-address]
```

- **Resets only a single neighbor.**
- **BGP session makes the transition from established to idle; everything from this neighbor must be relearned.**
- **Less severe than** clear ip bgp ***.**

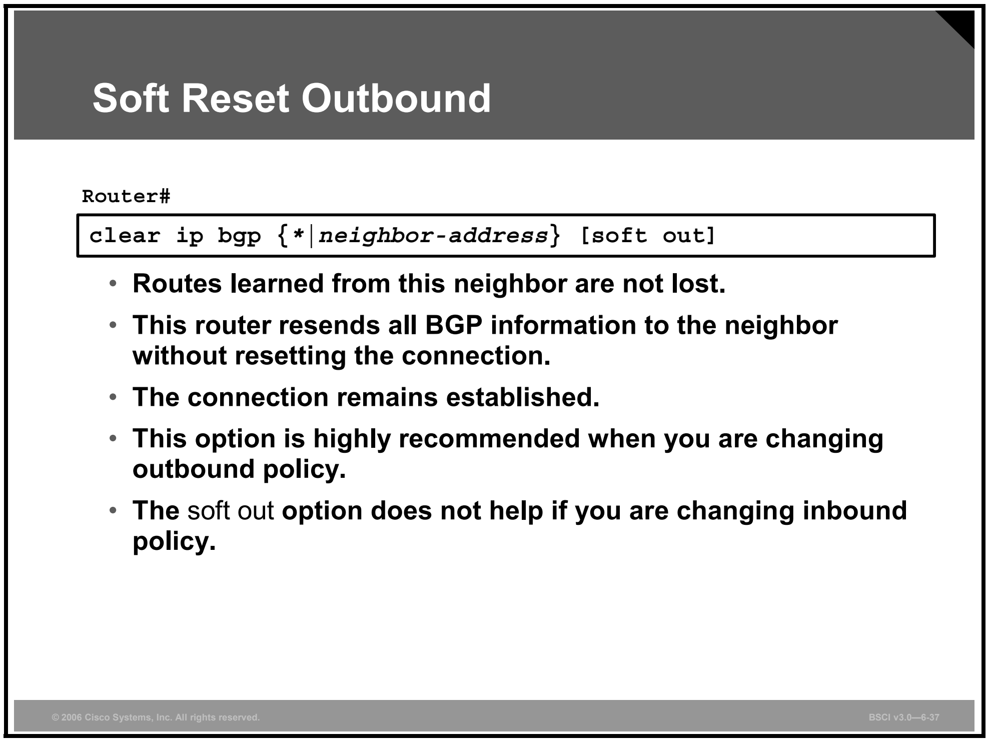
Resetting a session is a method of informing the neighbor or neighbors of a policy change. If BGP sessions are reset, all information received on those sessions is invalidated and removed from the BGP table. Also, the remote neighbor will detect a BGP session down state and will invalidate the routes that were received. After 30 to 60 seconds, the BGP sessions are reestablished automatically and the BGP table is exchanged again, but through the new filters. However, resetting the BGP session disrupts packet forwarding.

The two commands shown in the figure both cause a hard reset of the BGP neighbors that are involved. A hard reset means that the router issuing either of these commands will close the appropriate TCP connections, reestablish those TCP sessions as appropriate, and resend all information to each of the neighbors affected by the particular command that is used.

The **clear ip bgp *** command causes the BGP forwarding table on the router that issued this command to be completely deleted, and all networks must be relearned from every neighbor. If a router has multiple neighbors, this action is a very dramatic event. This command forces all neighbors to resend their entire tables simultaneously.

For example, consider a situation in which router A has eight neighbors and each neighbor has a full Internet table of about 32 MB in size. If router A issues the **clear ip bgp *** command, all eight routers resend their 32-MB tables at the same time. To hold all these updates, router A would need 256 MB of RAM. Router A would also need to be able to process all of this information. Processing 256 MB of updates would take a considerable number of CPU cycles for router A, further delaying the routing of user data.

If the second command, **clear ip bgp** [*neighbor-address*], is used instead, one neighbor is reset at a time. The impact is less severe on the router that is issuing this command; however, it takes longer to change policy for all of the neighbors, because each must be done individually rather than all at once as with the **clear ip bgp *** command. The **clear ip bgp** [*neighbor-address*] command still performs a hard reset and must reestablish the TCP session with the specified address, but this command affects only a single neighbor at a time.



Soft Reset Outbound

```
Router#
clear ip bgp {*|neighbor-address} [soft out]
```

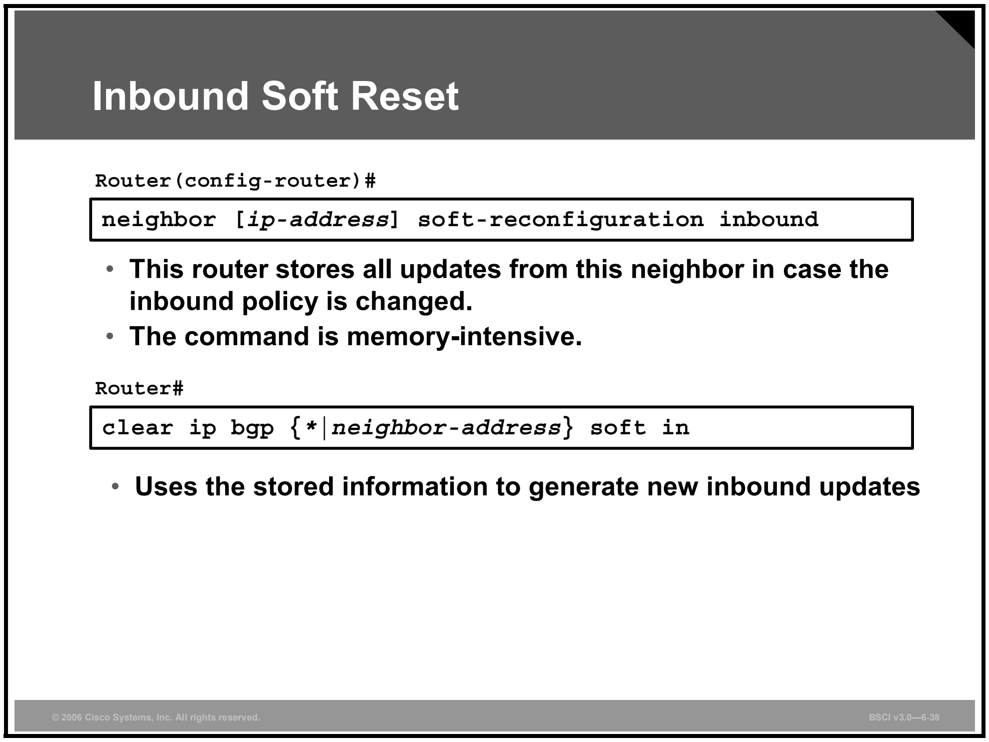
- **Routes learned from this neighbor are not lost.**
- **This router resends all BGP information to the neighbor without resetting the connection.**
- **The connection remains established.**
- **This option is highly recommended when you are changing outbound policy.**
- **The** soft out **option does not help if you are changing inbound policy.**

BSCI v3.0—6-37

The **soft out** option of the **clear ip bgp** command causes BGP to do a soft reset for outbound updates. The router issuing the **soft out** command does not reset the BGP session; instead, the router creates a new update and sends the whole table to the specified neighbors.

This update includes withdrawal commands for the networks that the other neighbor will not see anymore based on the new outbound policy.

Note The **soft** keyword of this command is optional; **clear ip bgp out** does a soft reset for outbound updates.



Inbound Soft Reset

Router(config-router)#

```
neighbor [ip-address] soft-reconfiguration inbound
```

- **This router stores all updates from this neighbor in case the inbound policy is changed.**
- **The command is memory-intensive.**

Router#

```
clear ip bgp {*|neighbor-address} soft in
```

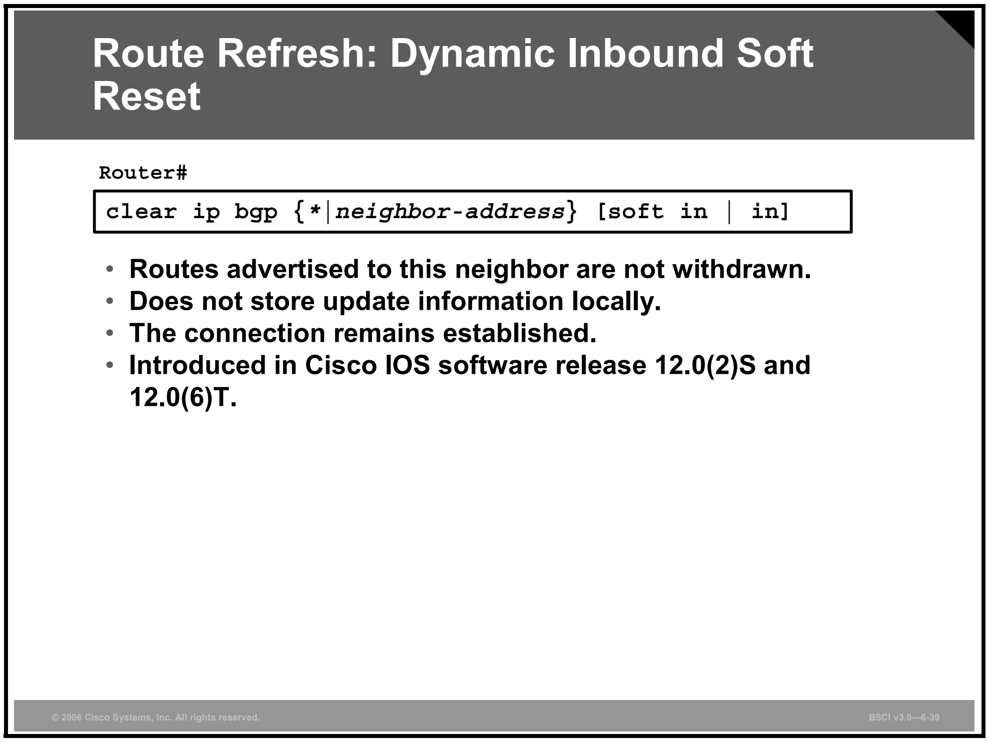
- **Uses the stored information to generate new inbound updates**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-38

There are two ways to perform an inbound soft reconfiguration: using stored routing update information. as shown in this figure, and dynamically, as shown in the next figure.

Enter the **neighbor** command that is shown in this figure to inform BGP to save all updates that were learned from the neighbor specified. The BGP router retains an unfiltered table of what that neighbor has sent.

When the inbound policy is changed, use the **clear ip bgp** command shown in the figure. The stored unfiltered table is used to generate new inbound updates; the new results are placed in the BGP forwarding database. Thus, if you make changes, you do not have to force the other side to resend everything.



Route Refresh: Dynamic Inbound Soft Reset

```
Router#
clear ip bgp {*|neighbor-address} [soft in | in]
```

- **Routes advertised to this neighbor are not withdrawn.**
- **Does not store update information locally.**
- **The connection remains established.**
- **Introduced in Cisco IOS software release 12.0(2)S and 12.0(6)T.**

BSCI v3.0—6-39

Cisco IOS Software Release 12.0(2)S and 12.0(6)T introduced a BGP soft reset enhancement feature, also known as route refresh, that provides automatic support for dynamic soft reset of inbound BGP routing table updates that is not dependent upon stored routing table update information. The **clear ip bgp soft in** command implements this feature. This method requires no preconfiguration and requires significantly less memory than the previous soft method for inbound routing table updates.

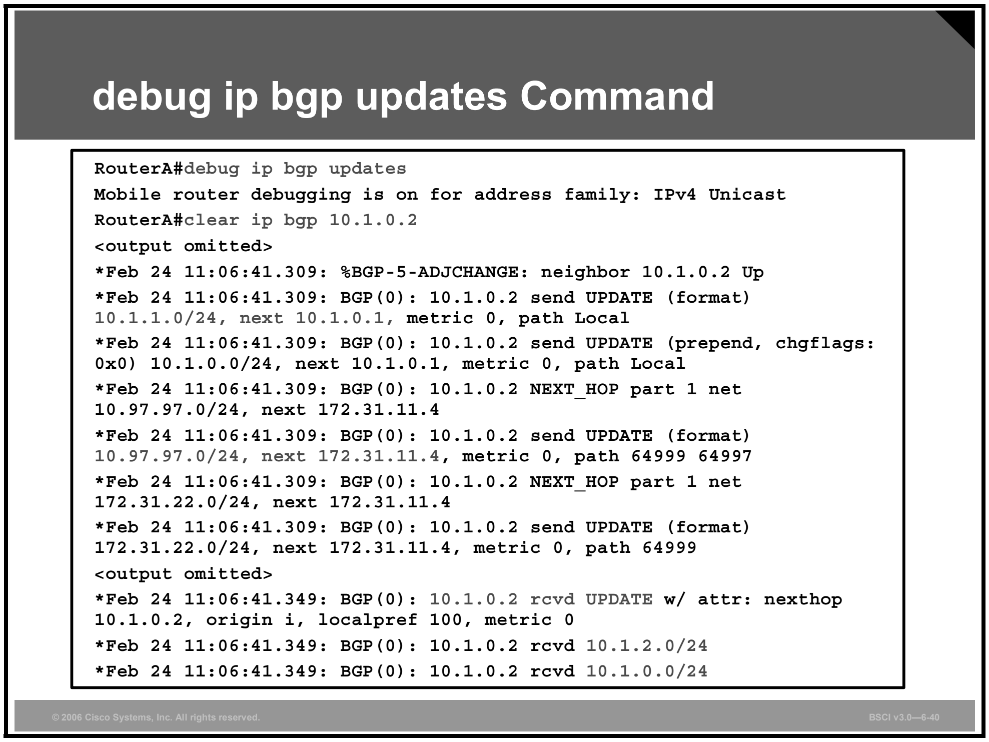
The **soft in** option generates new inbound updates without resetting the BGP session, but it can be memory-intensive. BGP does not allow a router to force another BGP speaker to resend its entire table. If you change the inbound BGP policy and you do not want to complete a hard reset, configure the router to perform a soft reconfiguration.

Note To determine whether a BGP router supports this route refresh capability, use the **show ip bgp neighbors** command. The following message is displayed in the output when the router supports the route refresh capability:

```
Received route refresh capability from peer.
```

If all BGP routers support the route refresh capability, use the **clear ip bgp** {***** | *address* | *peer-group-name*} **in** command. You need not use the **soft** keyword, because soft reset is automatically assumed when the route refresh capability is supported.

Note The **clear ip bgp soft** command performs a soft reconfiguration of both inbound and outbound updates.



debug ip bgp updates Command

```
RouterA#debug ip bgp updates
Mobile router debugging is on for address family: IPv4 Unicast
RouterA#clear ip bgp 10.1.0.2
<output omitted>
*Feb 24 11:06:41.309: %BGP-5-ADJCHANGE: neighbor 10.1.0.2 Up
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
10.1.1.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (prepend, chgflags:
0x0) 10.1.0.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net
10.97.97.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
10.97.97.0/24, next 172.31.11.4, metric 0, path 64999 64997
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net
172.31.22.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
172.31.22.0/24, next 172.31.11.4, metric 0, path 64999
<output omitted>
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd UPDATE w/ attr: nexthop
10.1.0.2, origin i, localpref 100, metric 0
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.2.0/24
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.0.0/24
```

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-40

Example: The debug ip bgp updates Command

The figure shows partial output from the **debug ip bgp updates** command on router A after the **clear ip bgp** command is issued to clear BGP sessions with its IBGP neighbor 10.1.0.2.

After the neighbor adjacency is reestablished, router A creates and sends updates to 10.1.0.2. The first update highlighted in the figure, 10.1.1.0/24, next 10.1.0.1, is an update about network 10.1.1.0/24, with a next hop of 10.1.0.1, which is the address of router A.

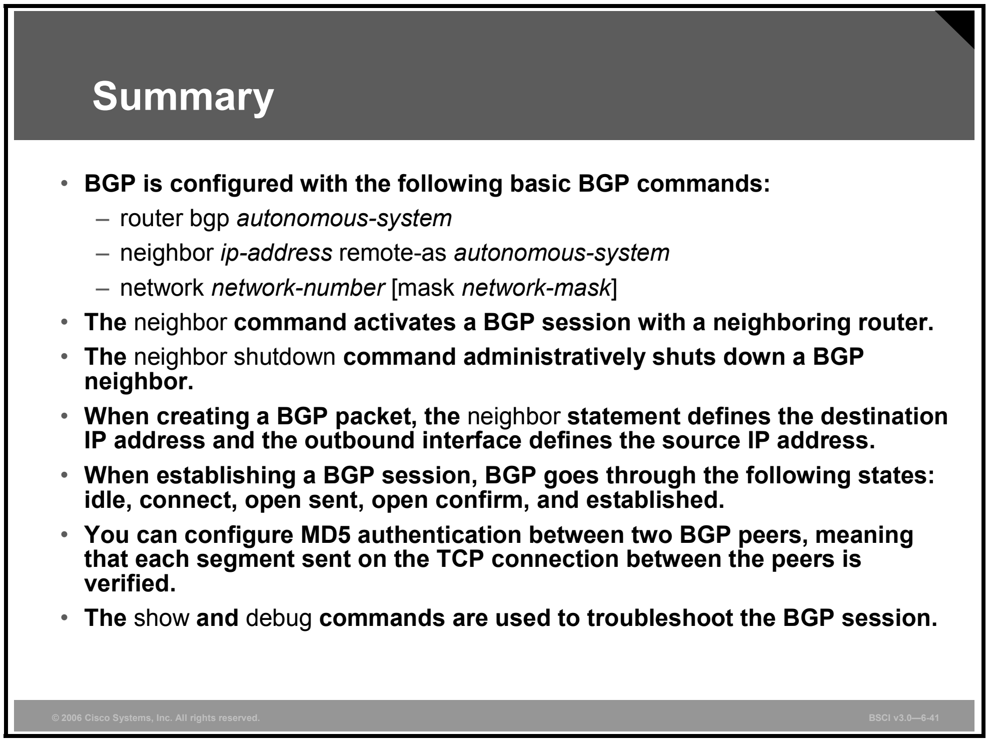
The second update highlighted in the figure, 10.97.97.0/24, next 172.31.11.4, is an update about network 10.97.97.0/24, with a next hop of 172.31.11.4, which is the address of one of the EBGP neighbors of router A. The EBGP next-hop address is being carried into IBGP.

Router A later receives updates from 10.1.0.2. The update highlighted in the figure contains a path to two networks, 10.1.2.0/24 and 10.1.0.0/24. The attributes shown in this update are described in the next lesson.

Note Debugging uses up router resources and should be turned on only when necessary.

Summary

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



Summary

- **BGP is configured with the following basic BGP commands:**
  - router bgp *autonomous-system*
  - neighbor *ip-address* remote-as *autonomous-system*
  - network *network-number* [mask *network-mask*]
- **The** neighbor **command activates a BGP session with a neighboring router.**
- **The** neighbor shutdown **command administratively shuts down a BGP neighbor.**
- **When creating a BGP packet, the** neighbor **statement defines the destination IP address and the outbound interface defines the source IP address.**
- **When establishing a BGP session, BGP goes through the following states: idle, connect, open sent, open confirm, and established.**
- **You can configure MD5 authentication between two BGP peers, meaning that each segment sent on the TCP connection between the peers is verified.**
- **The** show **and** debug **commands are used to troubleshoot the BGP session.**

Lesson 4

Selecting a BGP Path

Overview

Border Gateway Protocol (BGP) is used to perform policy-based routing. To manipulate the best paths chosen by BGP, a network administrator must understand the attributes that BGP uses and how BGP selects the best path based on these attributes.

This lesson explains the various BGP attributes, the characteristics of each, and how they are evaluated for BGP to select the best path to a given network.

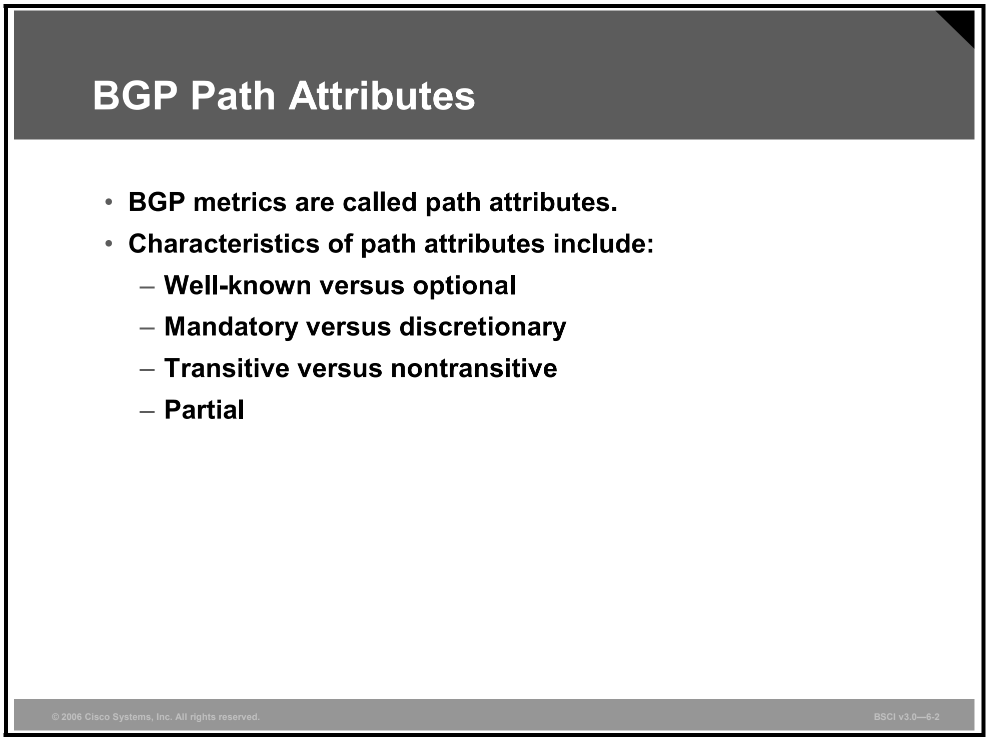
Objectives

Upon completing this lesson, you will be able to manipulate BGP path selection using BGP attributes. This ability includes being able to meet these objectives:

- Describe the characteristics of BGP attributes
- Describe the characteristics of the AS path attribute
- Describe the characteristics of the next-hop attribute
- Describe the characteristics of the origin attribute
- Describe the characteristics of the local preference attribute
- Describe the characteristics of the MED attribute
- Describe the characteristics of the weight attribute
- Describe the criteria for selecting a BGP path
- Describe how to select the best path to a destination network

Characteristics of BGP Attributes

BGP attributes inform BGP routers receiving updates about how to treat the paths to the final network. This topic describes the characteristics of BGP attributes.



BGP Path Attributes

- **BGP metrics are called path attributes.**
- **Characteristics of path attributes include:**
  - **Well-known versus optional**
  - **Mandatory versus discretionary**
  - **Transitive versus nontransitive**
  - **Partial**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-2

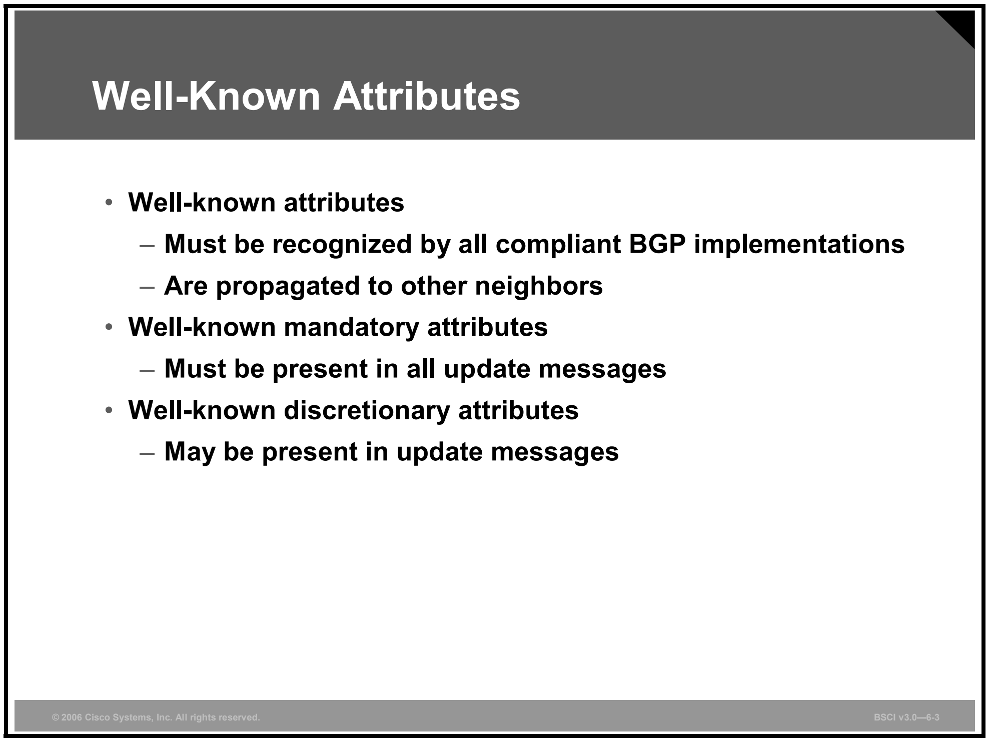
BGP routers send BGP update messages about destination networks to other BGP routers. The BGP update messages contain one or more routes and a set of BGP metrics, which are called path attributes, attached to the routes.

An attribute is either well-known or optional, mandatory or discretionary, and transitive or nontransitive. An attribute may also be partial.

Not all combinations of these characteristics are valid. Path attributes fall into the following four categories:

- Well-known mandatory
- Well-known discretionary
- Optional transitive
- Optional nontransitive

Only optional transitive attributes can be marked as partial.



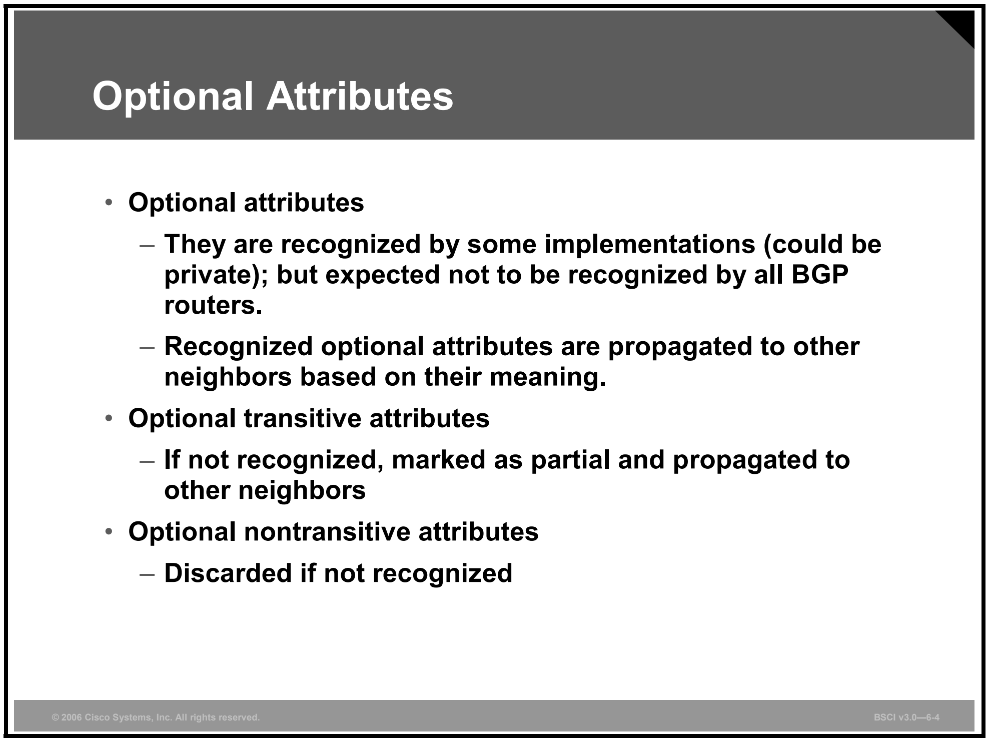
Well-Known Attributes

- Well-known attributes
  - Must be recognized by all compliant BGP implementations
  - Are propagated to other neighbors
- Well-known mandatory attributes
  - Must be present in all update messages
- Well-known discretionary attributes
  - May be present in update messages

BSCI v3.0—6-3

All BGP routers must recognize a well-known attribute and propagate it to the other BGP neighbors.

Well-known attributes are either mandatory or discretionary. A well-known mandatory attribute must be present in all BGP updates. A well-known discretionary attribute does not have to be present in all BGP updates.



Optional Attributes

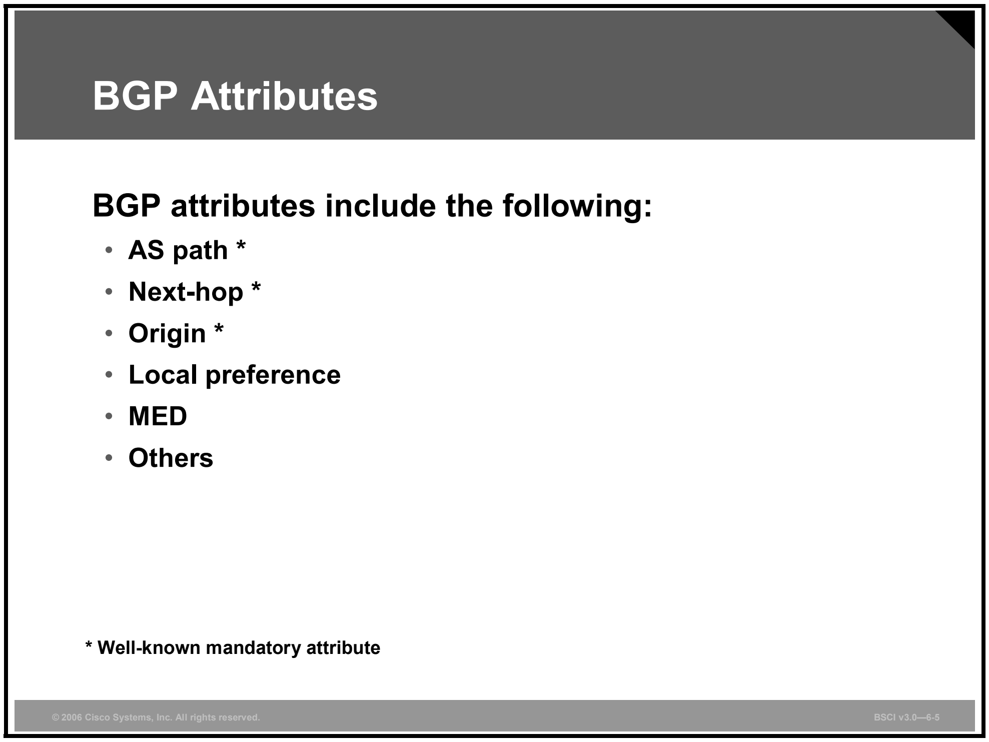
- **Optional attributes**
  - **They are recognized by some implementations (could be private); but expected not to be recognized by all BGP routers.**
  - **Recognized optional attributes are propagated to other neighbors based on their meaning.**
- **Optional transitive attributes**
  - **If not recognized, marked as partial and propagated to other neighbors**
- **Optional nontransitive attributes**
  - **Discarded if not recognized**

BSCI v3.0—6-4

Attributes that are not well-known are called optional. BGP routers do not have to support an optional attribute. Optional attributes are either transitive or nontransitive.

The following statements apply to optional attributes:

- BGP routers that implement the optional attribute may propagate it to the other BGP neighbors, based on its meaning.
- BGP routers that do not implement an optional transitive attribute should pass it to other BGP routers untouched and mark the attribute as partial.
- BGP routers that do not implement an optional nontransitive attribute must delete the attribute and must not pass it to other BGP routers.



BGP Attributes

BGP attributes include the following:

- **AS path ***
- **Next-hop ***
- **Origin ***
- **Local preference**
- **MED**
- **Others**

*** Well-known mandatory attribute**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-5

The following is a list of the common BGP attributes, by the categories that they belong to:

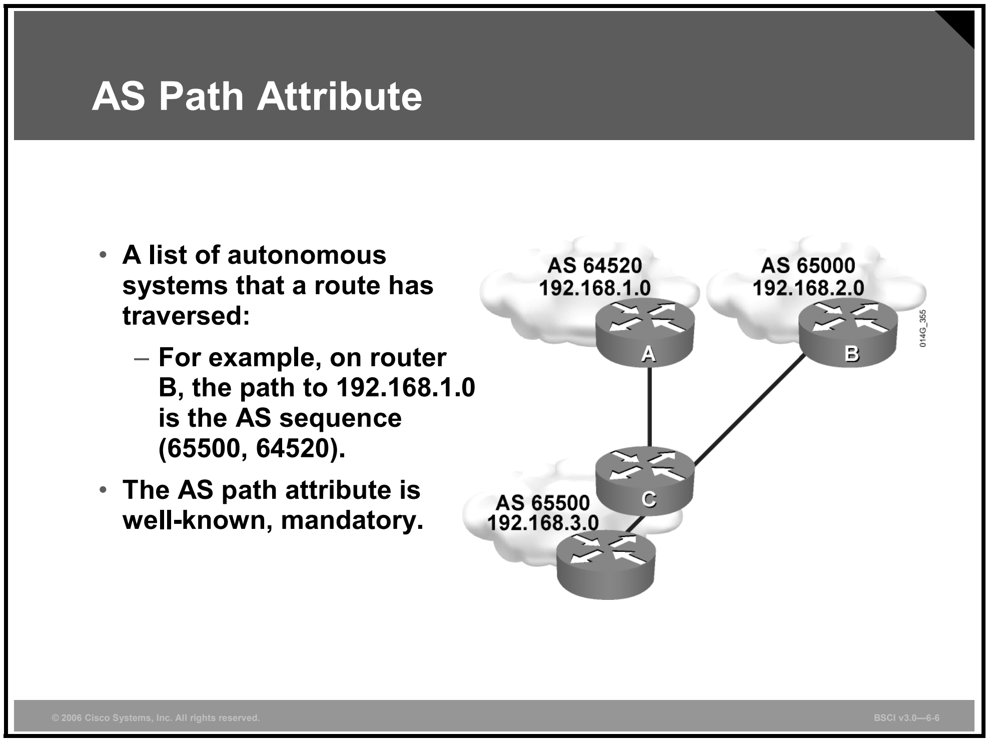
- Well-known mandatory attributes
  - Autonomous system (AS) path
  - Next-hop
  - Origin
- Well-known discretionary attributes
  - Local preference
  - Atomic aggregate
- Optional transitive attribute
  - Aggregator
- Optional nontransitive attribute
  - Multi-exit discriminator (MED)

Note In addition, Cisco uses a weight attribute for BGP. The weight attribute is an attribute that is defined by Cisco. The weight is configured locally on a router and is not propagated to any other BGP routers.

Note The attributes in this list are detailed in the following topics, except for the atomic aggregate and aggregator attributes. These two attributes relate to BGP summarization (or aggregation); more information on them can be found on Cisco.com.

AS Path Attribute

This topic describes the characteristics of the AS path attribute.



AS Path Attribute

- A list of autonomous systems that a route has traversed:
  - For example, on router B, the path to 192.168.1.0 is the AS sequence (65500, 64520).
- The AS path attribute is well-known, mandatory.

BSCI v3.0—6-6

The AS path is a well-known mandatory attribute. Whenever a route update passes through an AS, the AS number is prepended (added) to that update when it is advertised to the next External Border Gateway Protocol (EBGP) neighbor.

The AS path attribute is actually the list of AS numbers that a route has traversed to reach a destination, with the number of the AS that originated the route at the end of the list.

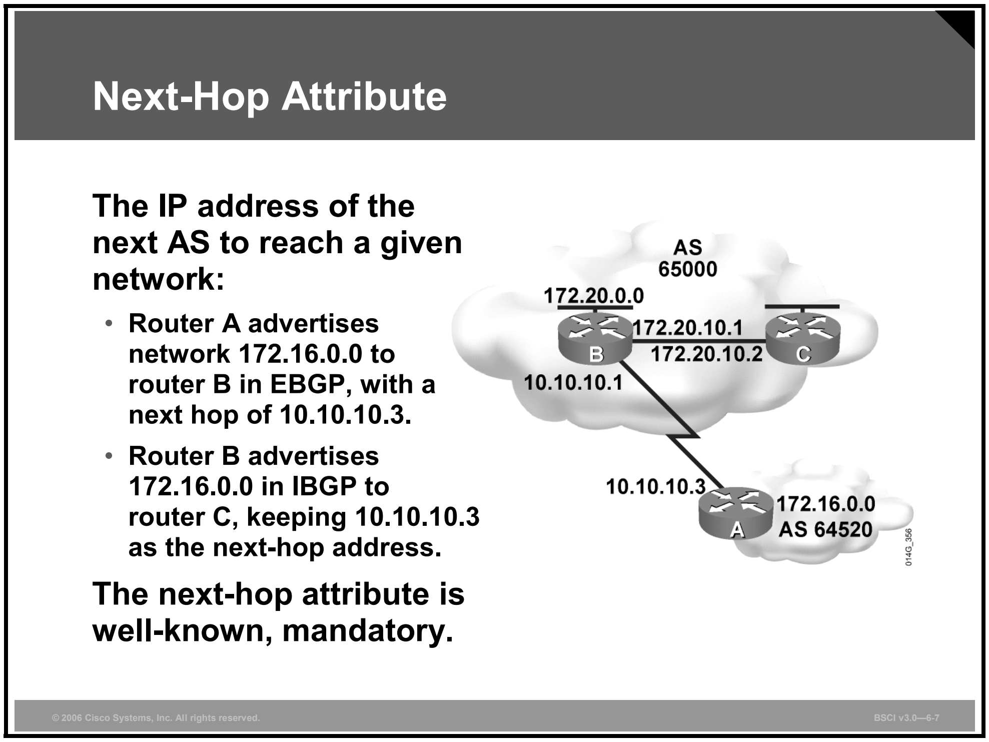
Example: AS Path Attribute

In the figure, router A in AS 64520 advertises network 192.168.1.0. When that route traverses AS 65500, router C prepends its own AS number to it. When 192.168.1.0 reaches router B, it has two AS numbers attached to it. From the perspective of router B, the path to reach 192.168.1.0 is (65500, 64520).

A similar process applies for the paths to networks 192.168.2.0 and 192.168.3.0. The path from router A to 192.168.2.0 is (65500, 65000), which means traverse AS 65500 and then AS 65000. Router C will have to traverse path (65000) to reach 192.168.2.0, and path (64520) to reach 192.168.1.0.

Next-Hop Attribute

This topic describes the characteristics of the next-hop attribute.



Next-Hop Attribute

The IP address of the next AS to reach a given network:

- **Router A advertises network 172.16.0.0 to router B in EBGP, with a next hop of 10.10.10.3.**
- **Router B advertises 172.16.0.0 in IBGP to router C, keeping 10.10.10.3 as the next-hop address.**

The next-hop attribute is well-known, mandatory.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-7

The BGP next-hop attribute is a well-known mandatory attribute that indicates the next-hop IP address that is to be used to reach a destination.

BGP routes AS by AS, not router by router. The next-hop address of a network from another AS will be an IP address of the entry point of the next AS along the path to that destination network.

Example: Next-Hop Attribute

For EBGP, the next hop is the IP address of the neighbor that sent the update. In the figure, router A will advertise 172.16.0.0 to router B, with a next hop of 10.10.10.3, and router B will advertise 172.20.0.0 to router A, with a next hop of 10.10.10.1.

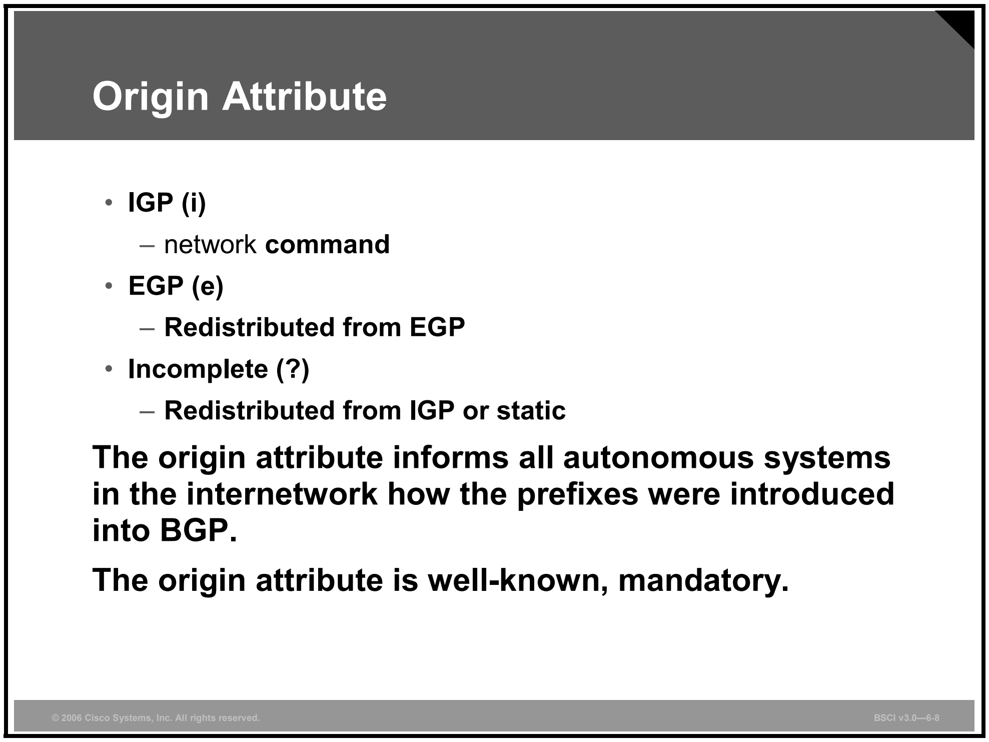
For Internal Border Gateway Protocol (IBGP), the protocol states that the next hop that is advertised by EBGP should be carried into IBGP. Because of that rule, router B will advertise 172.16.0.0 to its IBGP peer router C with a next hop of 10.10.10.3 (router A address); therefore, router C knows that the next hop to reach 172.16.0.0 is 10.10.10.3, not 172.20.10.1, as you might expect.

It is therefore very important that router C knows how to reach the 10.10.10.0 subnet, either via an interior gateway protocol (IGP) or a static route; otherwise, it will drop packets destined to 172.16.0.0 because it will not be able to get to the next-hop address for that network.

Alternatively, router B can change the next-hop attribute to itself if you issue the **neighbor next-hop-self** command.

Origin Attribute

This topic describes the characteristics of the origin attribute.



Origin Attribute

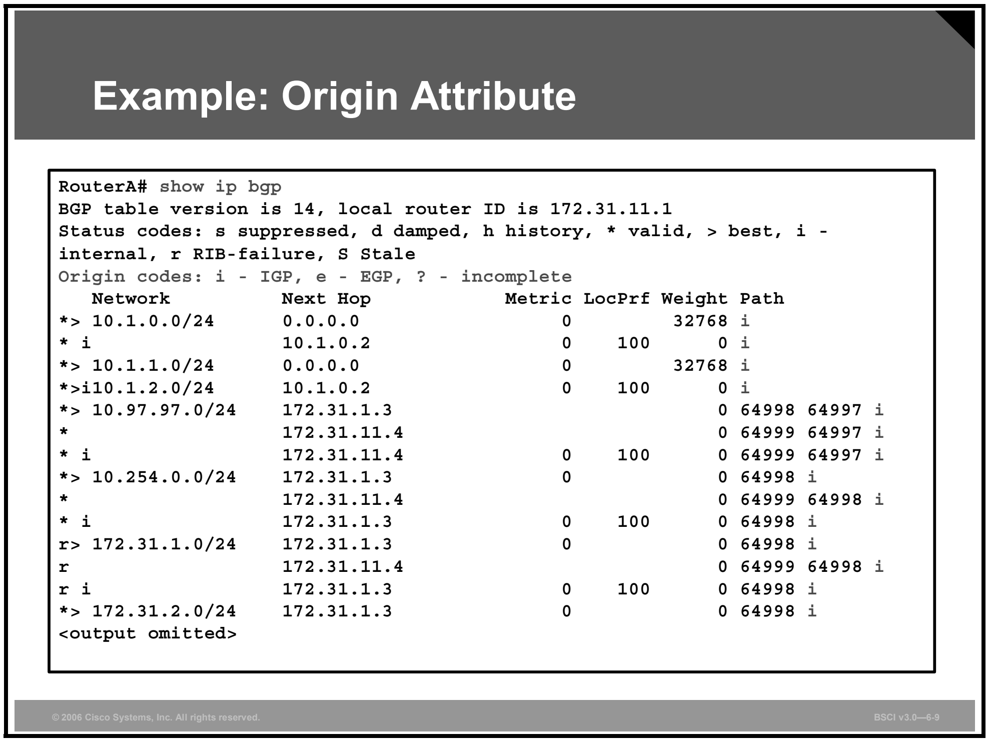
- **IGP (i)**
  - network **command**
- **EGP (e)**
  - **Redistributed from EGP**
- **Incomplete (?)**
  - **Redistributed from IGP or static**

The origin attribute informs all autonomous systems in the internetwork how the prefixes were introduced into BGP.

The origin attribute is well-known, mandatory.

The origin attribute defines the origin of the path information. The origin attribute can be one of these three values:

- **IGP:** The route is interior to the originating AS. This value normally results when the **network** command is used to advertise the route via BGP. An origin of IGP is indicated with an “i” in the BGP table.
- **Exterior Gateway Protocol (EGP):** The route has been learned via EGP. This value is indicated with an “e” in the BGP table. EGP is considered a historical routing protocol and is not supported on the Internet because it performs only classful routing and does not support classless interdomain routing (CIDR).
- **Incomplete:** The origin of the route is unknown or has been learned by some other means. This value usually results when a route is redistributed into BGP. An incomplete origin is indicated with a question mark (?) in the BGP table.



Example: Origin Attribute

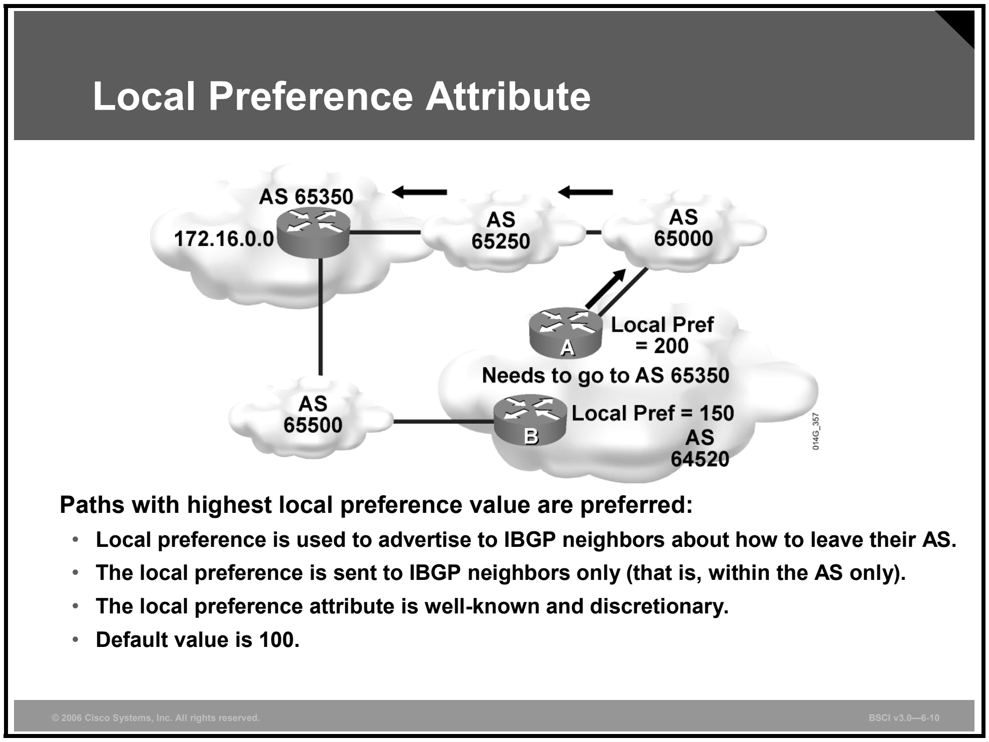
```
RouterA# show ip bgp
BGP table version is 14, local router ID is 172.31.11.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal, r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 10.1.0.0/24      0.0.0.0                  0         32768 i
* i                 10.1.0.2                 0    100      0 i
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*>i10.1.2.0/24      10.1.0.2                 0    100      0 i
*> 10.97.97.0/24    172.31.1.3                             0 64998 64997 i
*                   172.31.11.4                            0 64999 64997 i
* i                 172.31.11.4              0    100      0 64999 64997 i
*> 10.254.0.0/24    172.31.1.3               0             0 64998 i
*                   172.31.11.4                            0 64999 64998 i
* i                 172.31.1.3               0    100      0 64998 i
r> 172.31.1.0/24    172.31.1.3               0             0 64998 i
r                   172.31.11.4                            0 64999 64998 i
r i                 172.31.1.3               0    100      0 64998 i
*> 172.31.2.0/24    172.31.1.3               0             0 64998 i
<output omitted>
```

Example: Origin Attribute

The figure shows an example output of the **show ip bgp** command. The origin code, reflecting the origin attribute, is in the last column, at the end of each line. In this example, all origin codes are “i,” indicating an origin attribute of IGP; the routes are interior to the originating AS.

Local Preference Attribute

This topic describes the characteristics of the local preference attribute.



Local Preference Attribute

Paths with highest local preference value are preferred:

- **Local preference is used to advertise to IBGP neighbors about how to leave their AS.**
- **The local preference is sent to IBGP neighbors only (that is, within the AS only).**
- **The local preference attribute is well-known and discretionary.**
- **Default value is 100.**

Local preference is a well-known discretionary attribute that provides an indication to routers in the AS about which path is preferred to exit the AS. A path with a higher local preference is preferred.

The local preference is an attribute that is configured on a router and exchanged among routers within the same AS only. The default value for local preference on a Cisco router is 100.

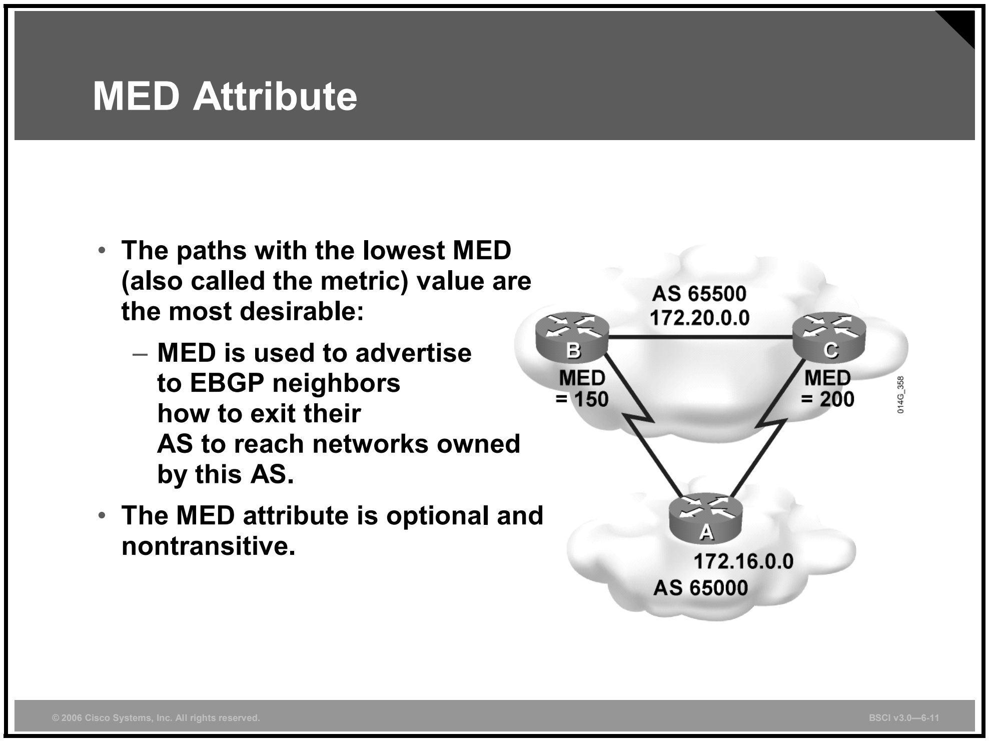
Example: Local Preference Attribute

In the figure, AS 64520 receives updates about network 172.16.0.0 from two directions. The local preference on router A for network 172.16.0.0 is set to 200, and the local preference on router B for network 172.16.0.0 is set to 150.

Because the local preference information is exchanged within AS 64520, all traffic in AS 64520 addressed to network 172.16.0.0 will be sent to router A as an exit point from AS 64520 (because of its higher local preference).

MED Attribute

This topic describes the characteristics of the MED attribute.



MED Attribute

- The paths with the lowest MED (also called the metric) value are the most desirable:
  - MED is used to advertise to EBGP neighbors how to exit their AS to reach networks owned by this AS.
- The MED attribute is optional and nontransitive.

BSCI v3.0—6-11

The MED attribute, also called the metric, is an optional nontransitive attribute.

The MED is an indication to EBGP neighbors about the preferred path into an AS. The MED attribute is a dynamic way to influence another AS about which path that it should choose to reach a certain route when multiple entry points into an AS exist. A lower metric is preferred.

Unlike local preference, the MED is exchanged between autonomous systems. The MED is sent to EBGP peers; those routers propagate the MED within their AS, and the routers within the AS use the MED but do not pass it on to the next AS. When the same update is passed on to another AS, the metric is set back to the default of 0.

MED influences inbound traffic to an AS, and local preference influences outbound traffic from an AS.

By default, a router compares the MED attribute only for paths from neighbors in the same AS.

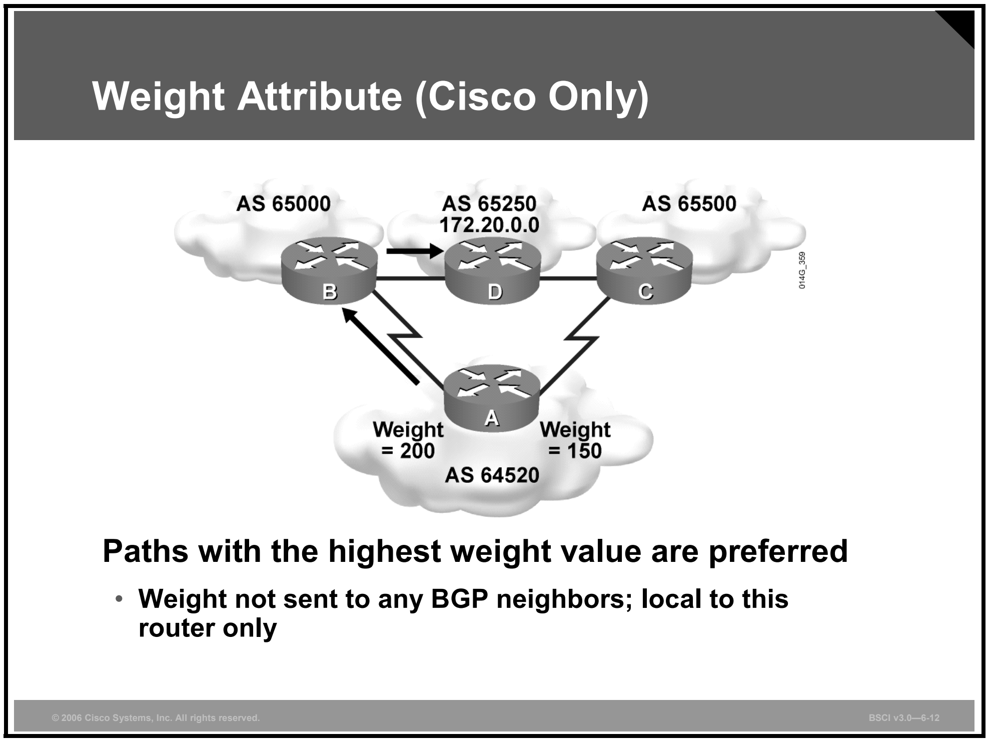
Note The MED attribute means that BGP is the only protocol that can affect how routes are sent into an AS.

Example: MED Attribute

In the figure, the router B MED attribute is set to 150, and the router C MED attribute is set to 200. When router A receives updates from routers B and C, it picks router B as the best next hop because its MED of 150 is less than the router C MED of 200.

Weight Attribute

This topic describes the characteristics of the weight attribute.



Weight Attribute (Cisco Only)

Paths with the highest weight value are preferred

- **Weight not sent to any BGP neighbors; local to this router only**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-12

The weight attribute is an attribute that Cisco defines for the path selection process. The weight is configured locally on a router and is not propagated to any other routers. This attribute applies when you are using one router with multiple exit points out of an AS, as opposed to the local preference attribute, which is used when two or more routers provide multiple exit points.

The weight can have a value from 0 to 65535. Paths that the router originates have a weight of 32768 by default, and other paths have a weight of 0 by default.

Routes with a higher weight are preferred when multiple routes exist to the same destination.

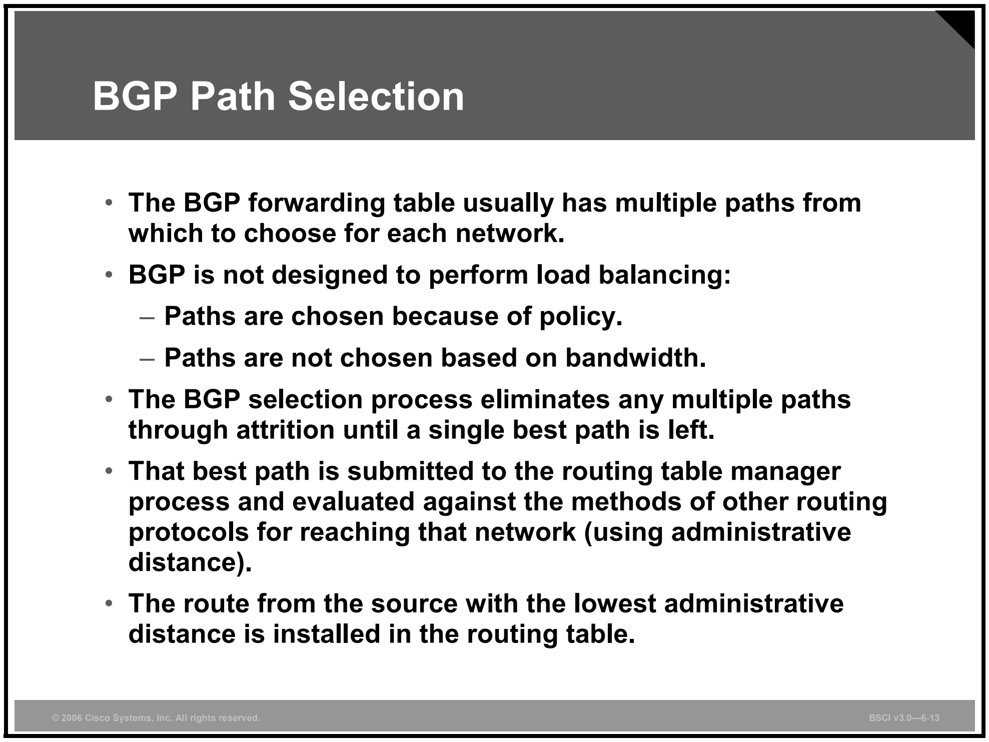
Example: Weight Attribute (Cisco Only)

In the figure, routers B and C learn about network 172.20.0.0 from AS 65250 and propagate the update to router A. Router A has two ways to reach 172.20.0.0, and it has to decide which route to take.

In the example, router A sets the weight of updates coming from router B to 200 and the weight of those coming from router C to 150. Because the weight for router B is higher than the weight for router C, router A uses router B as a next hop to reach 172.20.0.0.

Determining the BGP Path Selection

This topic describes the criteria for selecting a BGP path.



BGP Path Selection

- The BGP forwarding table usually has multiple paths from which to choose for each network.
- BGP is not designed to perform load balancing:
  - Paths are chosen because of policy.
  - Paths are not chosen based on bandwidth.
- The BGP selection process eliminates any multiple paths through attrition until a single best path is left.
- That best path is submitted to the routing table manager process and evaluated against the methods of other routing protocols for reaching that network (using administrative distance).
- The route from the source with the lowest administrative distance is installed in the routing table.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-13

Multiple paths may exist to reach a given network. As paths for the network are evaluated, those determined not to be the best path are eliminated from the selection criteria but kept in the BGP forwarding table (which can be displayed using the **show ip bgp** command) in case the best path becomes inaccessible.

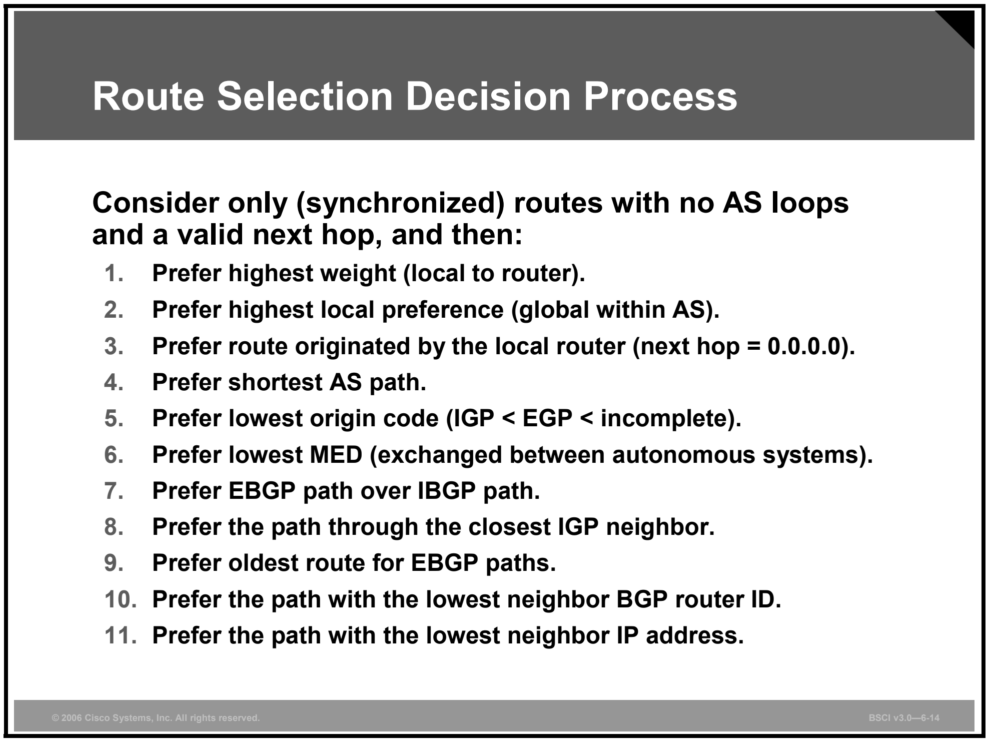
BGP is not designed to perform load balancing; paths are chosen because of policy, not based on bandwidth. The BGP selection process eliminates any multiple paths until a single best path is left.

The best path is submitted to the routing table manager process and is evaluated against any other routing protocols that can also reach that network. The route from the source with the lowest administrative distance is installed in the routing table.

The decision process is based on the attributes described earlier.

Selecting a BGP Path

This topic explains how the best path to a destination network is selected.



Route Selection Decision Process

Consider only (synchronized) routes with no AS loops and a valid next hop, and then:

1. **Prefer highest weight (local to router).**
2. **Prefer highest local preference (global within AS).**
3. **Prefer route originated by the local router (next hop = 0.0.0.0).**
4. **Prefer shortest AS path.**
5. **Prefer lowest origin code (IGP < EGP < incomplete).**
6. **Prefer lowest MED (exchanged between autonomous systems).**
7. **Prefer EBGP path over IBGP path.**
8. **Prefer the path through the closest IGP neighbor.**
9. **Prefer oldest route for EBGP paths.**
10. **Prefer the path with the lowest neighbor BGP router ID.**
11. **Prefer the path with the lowest neighbor IP address.**

BSCI v3.0—6-14

After BGP receives updates about different destinations from different autonomous systems, it decides the best path to choose to reach a specific destination. BGP chooses only a single best path to reach a destination.

The decision process is based on the BGP attributes. When faced with multiple routes to the same destination, BGP chooses the best route for routing traffic toward the destination. BGP considers only (synchronized) routes with no AS loops and a valid next hop. The following process summarizes how BGP chooses the best route on a Cisco router:

1. Prefer the route with the highest weight. (Recall that the weight is proprietary to Cisco and is local to the router only.)

2. If multiple routes have the same weight, prefer the route with the highest local preference. (Recall that the local preference is used within an AS.)

3. If multiple routes have the same local preference, prefer the route that the local router originated. A locally originated route has a next hop of 0.0.0.0 in the BGP table.

4. If none of the routes were locally originated, prefer the route with the shortest AS path.

5. If the AS path length is the same, prefer the lowest origin code (IGP < EGP < incomplete).

6. If all origin codes are the same, prefer the path with the lowest MED. (Recall that the MED is exchanged between autonomous systems.)

   The MED comparison is made only if the neighboring AS is the same for all routes considered, unless the **bgp always-compare-med** command is enabled.

   Note The most recent Internet Engineering Task Force (IETF) decision regarding BGP MED assigns a value of infinity to the missing MED, making the route lacking the MED variable the least preferred. The default behavior of BGP routers running Cisco IOS software is to treat routes without the MED attribute as having a MED of 0, making the route lacking the MED variable the most preferred. To configure the router to conform to the IETF standard, use the **bgp bestpath missing-as-worst** command.

7. If the routes have the same MED, prefer external paths (EBGP) to internal paths (IBGP).
8. If synchronization is disabled and only internal paths remain, prefer the path through the closest IGP neighbor. This step means that the router will prefer the shortest internal path within the AS to reach the destination (the shortest path to the BGP next hop).
9. For EBGP paths, select the oldest route to minimize the effect of routes going up and down (flapping).
10. Prefer the route with the lowest neighbor BGP router ID value.
11. If the BGP router IDs are the same, prefer the router with the lowest neighbor IP address.

Only the best path is entered in the routing table and propagated to the BGP neighbors of the router.

Note The route selection process summarized here does not cover all cases but is sufficient for a basic understanding of how BGP selects routes.

For example, suppose there are seven paths to reach network 10.0.0.0. All paths have no AS loops and have valid next-hop addresses, so all seven paths proceed to Step 1, which examines the weight of the paths.

All seven paths have a weight of 0, so they all proceed to Step 2, which examines the local preference of the paths. Four of the paths have a local preference of 200, and the other three have local preferences of 100, 100, and 150.

The four with a local preference of 200 will continue the evaluation process to the next step. The other three will still be in the BGP forwarding table, but are currently disqualified as the best path.

BGP will continue the evaluation process until only a single best path remains. The single best path that remains will be submitted to the IP routing table as the best BGP path.

Path Selection with Multihomed Connection

An AS rarely implements BGP with only one EBGP connection. This situation generally means that multiple paths exist for each network in the BGP forwarding database.

If only one path exists, if it is loop-free and synchronized with the IGP for IBGP, and if the next hop is reachable, the path is submitted to the IP routing table. There is no path selection taking place because there is only one path and manipulating it will produce no benefit.

Only the best path is put in the routing table and propagated to the BGP neighbors of the router.

Without route manipulation, the most common reason for path selection is Step 4, the preference for the shortest AS path.

Step 1 looks at weight, which by default is set to 0 for routes that were not originated by this router.

Step 2 compares local preference, which by default is set to 100 for all networks. Both of these steps have an effect only if the network administrator configures the weight or local preference to a nondefault value.

Step 3 looks at networks that are owned by this AS. If one of the routes is injected into the BGP table by the local router, the local router prefers it to any routes received from other BGP routers.

Step 4 selects the path that has the fewest autonomous systems to cross. This is the most common reason a path is selected in BGP. If a network administrator does not like the path with the fewest autonomous systems, the administrator needs to manipulate the weight or local preference to change which outbound path BGP chooses.

Step 5 looks at how a network was introduced into BGP. This introduction is usually accomplished either with network statements (i for an origin code) or through redistribution (? for an origin code).

Step 6 looks at MED to judge where the neighbor AS wants this AS to send packets for a given network. Cisco sets the MED to 0 by default; therefore, MED does not participate in path selection unless the network administrator of the neighbor AS manipulates the paths using MED.

If multiple paths have the same number of autonomous systems to traverse, the second most common decision point is Step 7, which states that an externally learned path from an EBGP neighbor is preferred over a path learned from an IBGP neighbor. A router in an AS prefers to use the Internet service provider (ISP) bandwidth to reach a network rather than using internal bandwidth to reach an IBGP neighbor on the other side of its own AS.

If the AS path is equal and the router in an AS has no EBGP neighbors for that network (only IBGP neighbors), it makes sense to take the quickest path to the nearest exit point. Step 8 looks for the closest IBGP neighbor. The IGP metric determines what "closest" means; for example, Routing Information Protocol (RIP) uses hop count, and Open Shortest Path First Protocol (OSPF) uses the least cost, based on bandwidth.

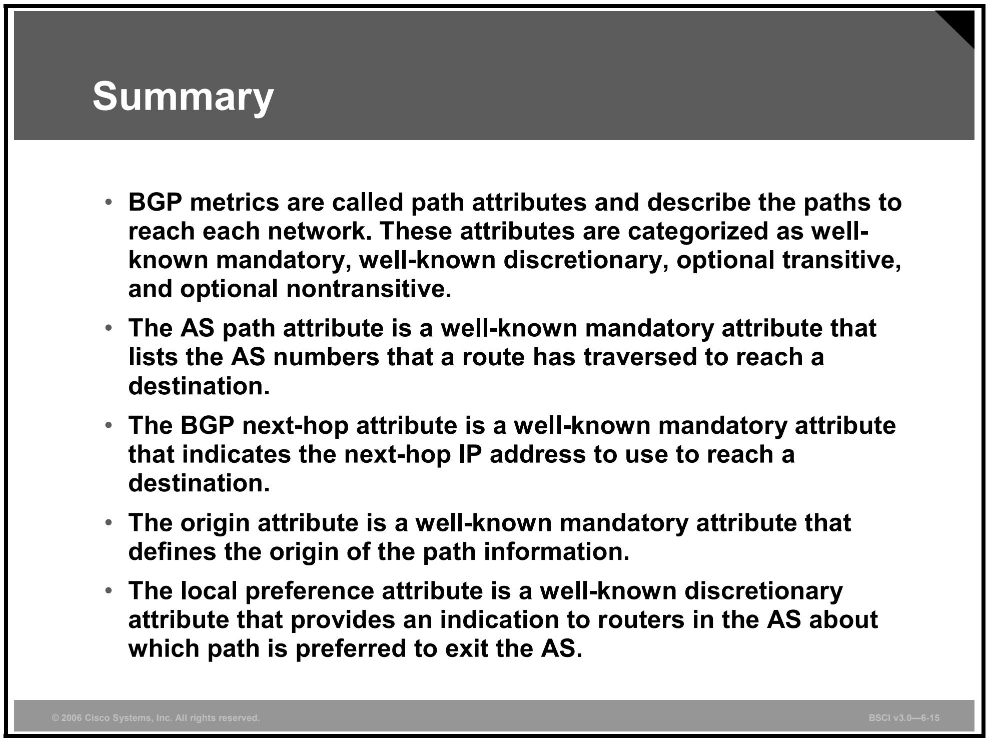
If the AS path is equal and the costs via all IBGP neighbors are equal, or if all neighbors for this network are EBGP, the oldest path, Step 9, is the next most common reason for selecting one path over another. EBGP neighbors rarely establish sessions at the exact same time. One session is likely to be older than another, so the paths through that older neighbor are considered more stable, because they have been up longer.

If all of the listed criteria are equal, the next most common decision is to take the neighbor with the lowest BGP router ID, which is Step 10.

If the BGP router IDs are the same (for example, if the paths are to the same BGP router), Step 11 states that the route with the lowest neighbor IP address is used.

Summary

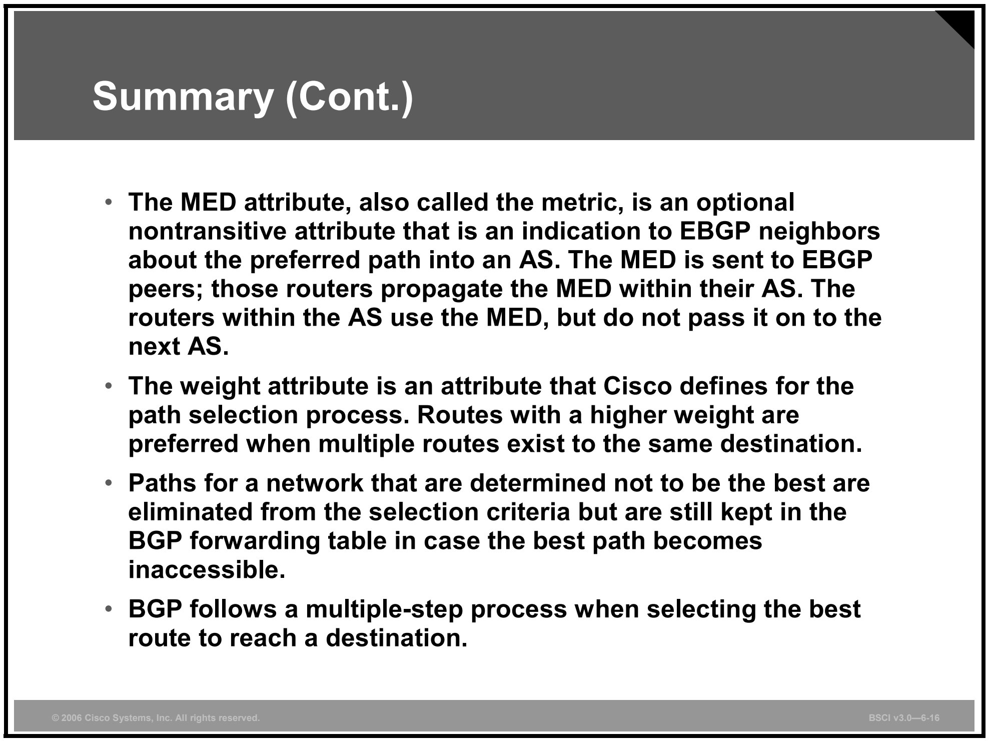
This topic summarizes the key points that were discussed in this lesson.



Summary

- BGP metrics are called path attributes and describe the paths to reach each network. These attributes are categorized as well-known mandatory, well-known discretionary, optional transitive, and optional nontransitive.
- The AS path attribute is a well-known mandatory attribute that lists the AS numbers that a route has traversed to reach a destination.
- The BGP next-hop attribute is a well-known mandatory attribute that indicates the next-hop IP address to use to reach a destination.
- The origin attribute is a well-known mandatory attribute that defines the origin of the path information.
- The local preference attribute is a well-known discretionary attribute that provides an indication to routers in the AS about which path is preferred to exit the AS.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-15



Summary (Cont.)

- The MED attribute, also called the metric, is an optional nontransitive attribute that is an indication to EBGP neighbors about the preferred path into an AS. The MED is sent to EBGP peers; those routers propagate the MED within their AS. The routers within the AS use the MED, but do not pass it on to the next AS.
- The weight attribute is an attribute that Cisco defines for the path selection process. Routes with a higher weight are preferred when multiple routes exist to the same destination.
- Paths for a network that are determined not to be the best are eliminated from the selection criteria but are still kept in the BGP forwarding table in case the best path becomes inaccessible.
- BGP follows a multiple-step process when selecting the best route to reach a destination.

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-16

Lesson 5

Using Route Maps to Manipulate Basic BGP Paths

Overview

This lesson discusses how to configure an autonomous system (AS) using route maps to manipulate the Border Gateway Protocol (BGP) local preference and multi-exit discriminator (MED) attributes to influence BGP path selection. The lesson concludes with a brief discussion of a typical enterprise BGP implementation, to tie together the concepts presented in this module.

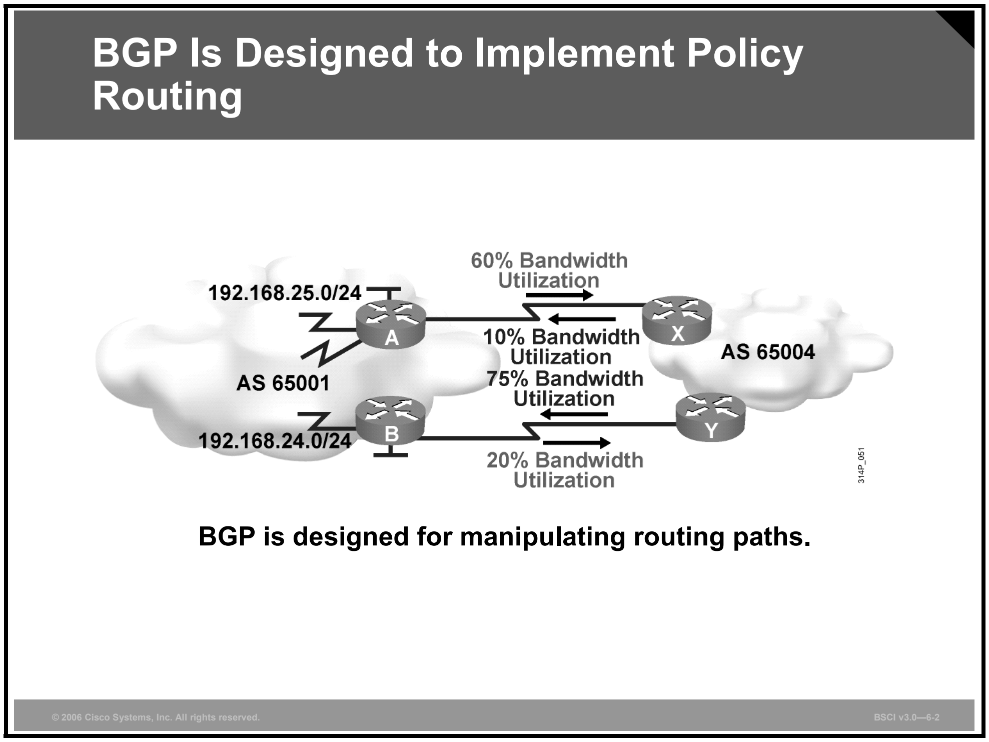
Objectives

Upon completing this lesson, you will be able to manipulate BGP traffic using route maps. This ability includes being able to meet these objectives:

- Describe how to set local preference with route maps
- Describe how to use route maps to set the BGP MED attribute
- Describe a typical enterprise BGP implementation

Setting Local Preference with Route Maps

This topic describes how to use route maps to set the BGP local preference attribute.



BGP Is Designed to Implement Policy Routing

60% Bandwidth Utilization
192.168.25.0/24
A
X
10% Bandwidth Utilization
AS 65004
AS 65001
75% Bandwidth Utilization
Y
B
192.168.24.0/24
20% Bandwidth Utilization
314P_051

BGP is designed for manipulating routing paths.

BSCI v3.0—6-2

Unlike local routing protocols, BGP was never designed to choose the quickest path. BGP was designed to manipulate traffic flow to maximize or minimize bandwidth use. This figure demonstrates a common situation that can result when you are using BGP without any policy manipulation.

Using default settings for path selection in BGP can cause uneven use of bandwidth. In the figure, router A in AS 65001 is using 60 percent of its outbound bandwidth to router X in 65004, but router B is using only 20 percent of its outbound bandwidth. If this utilization is acceptable to the administrator, then no manipulation is needed.

But if the load averages 60 percent and has temporary bursts above 100 percent of the bandwidth, this situation will cause lost packets, higher latency, and higher CPU usage because of the number of packets being routed. When another link to the same locations is available and is not heavily used, it makes sense to divert some of the traffic to the other path. To change outbound path selection from AS 65001, the network administrator must manipulate the local preference attribute.

To determine which path to manipulate, the administrator performs a traffic analysis on Internet-bound traffic by examining the most heavily visited addresses, web pages, or domain names. This information can usually be found by examining network management records or firewall accounting information.

Example: BGP Is Designed to Implement Policy Routing

In the example in this figure, assume that 35 percent of all traffic from AS 65001 has been going to www.cisco.com. The administrator can obtain the Cisco address or AS number by performing a reverse Domain Name System (DNS) lookup or by going to www.arin.net and looking up the AS number of Cisco Systems or the address space that is assigned to the company. After this information has been determined, the administrator uses local preference and route maps to manipulate path selection for the Cisco network.

Using a route map, router B can announce all networks that are associated with that AS with a higher local preference than router A announces for those networks. Other routers in AS 65001 running BGP will prefer the routes with the highest local preference. For the Cisco networks, router B announces the highest local preference, so all traffic destined for that AS will exit AS 65001 via router B. The outbound load for router B increases from its previous load of 20 percent to account for the extra traffic from AS 65001 destined for Cisco networks. The outbound load for router A, which was originally 60 percent, should decrease, and this change will bring the outbound load on both links into relative balance.

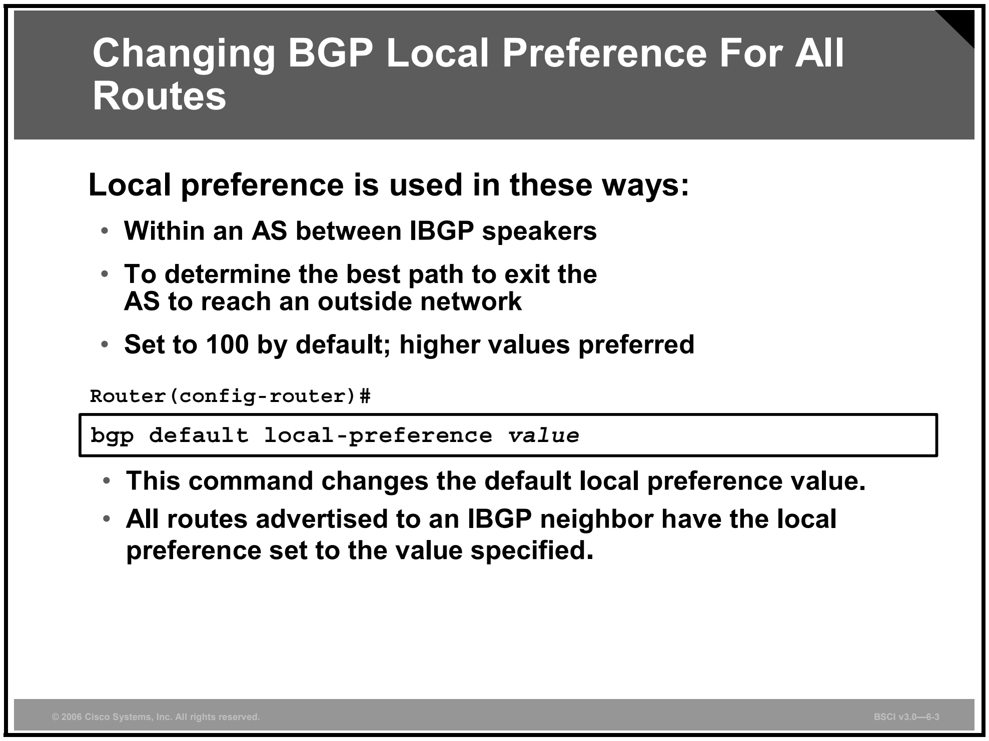
Just as there was a loading issue outbound from AS 65001, there can be a similar problem inbound. Maybe the sales web servers are located on the same subnet behind router B, causing the inbound load for router B to average higher utilization. To manipulate how traffic enters an AS, use the BGP MED attribute.

For example, AS 65001 announces a lower MED for network 192.168.25.0/24 to AS 65004 out router A. This MED is a recommendation to the next AS on how to enter AS 65001; however, the MED is not considered until Step 6 of the BGP path selection process. If AS 65004 prefers to keep its AS path via router Y to router B in AS 65001, then AS 65004 simply needs to have router Y announce a higher local preference to the BGP routers in AS 65004 for network 192.168.25.0/24 than router X announces. The local preference that router Y advertises to other BGP routers in AS 65004 is evaluated before the MED coming from router A in AS 65001. MED is considered a recommendation because the receiving AS can override it by having that AS manipulate a value before the MED is considered.

In the figure, assume that 55 percent of all traffic is going to the 192.168.25.0/24 subnet (router A). The inbound utilization to router A is averaging only 10 percent, but the inbound utilization to router B is averaging 75 percent. If AS 65001 were set to prefer to have all traffic going to 192.168.25.0/24 enter through router A from AS 65004, the load inbound on router A would increase, and the load inbound on router B would decrease.

The problem is that if the inbound load for router A spikes to more than 100 percent and causes the link to flap, all the sessions crossing that link could be lost. If these sessions were purchases being made on AS 65001 web servers, revenue would be lost, which is a result that administrators want to avoid.

If the load averages below 50 percent for the outbound or inbound case, path manipulation might not be needed; however, when a link starts to reach the capacity of the link for an extended period of time, more bandwidth is needed or path manipulation should be considered.



Changing BGP Local Preference For All Routes

Local preference is used in these ways:

- **Within an AS between IBGP speakers**
- **To determine the best path to exit the AS to reach an outside network**
- **Set to 100 by default; higher values preferred**

```
Router(config-router)#
bgp default local-preference value
```

- **This command changes the default local preference value.**
- **All routes advertised to an IBGP neighbor have the local preference set to the value specified.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-3

Local preference is used only within an AS between Internal Border Gateway Protocol (IBGP) speakers to determine the best path to leave the AS to reach an outside network.

The local preference is set to 100 by default; higher values are preferred.

The **bgp default local-preference** command changes the default local preference value.

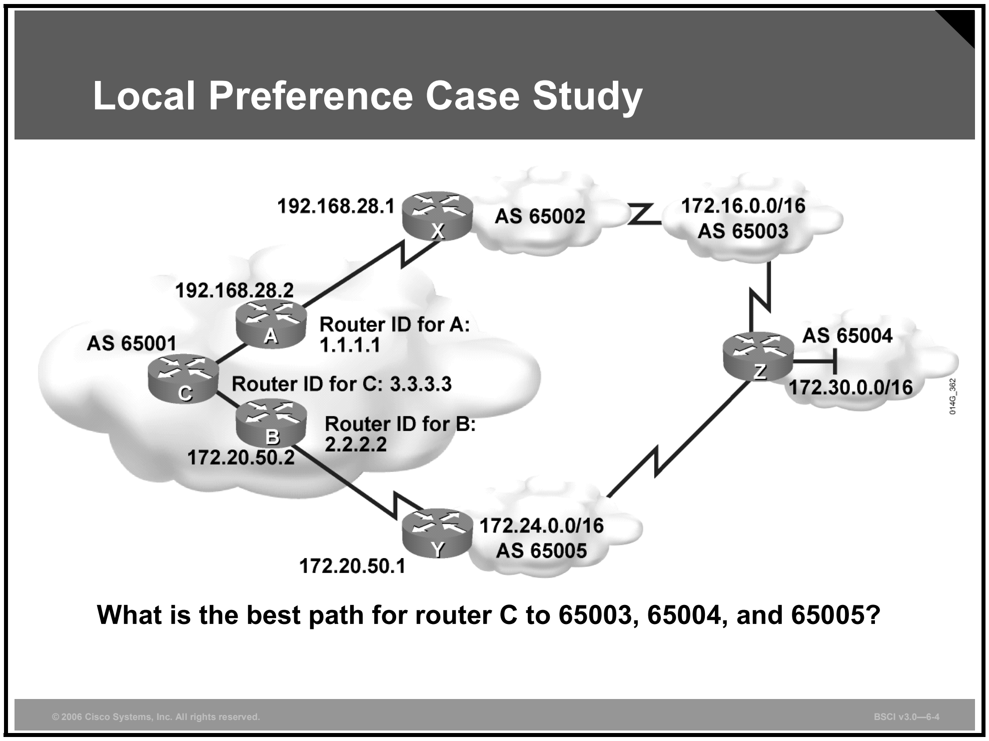
The bgp default local-preference Command Parameter

The table describes the syntax of the **bgp default local-preference** command.

Parameter	Description
value	Local preference value from 0 to 4294967295. A higher value is more preferred.

With this command, all IBGP routes that are advertised have the local preference set to the value specified.

If an External Border Gateway Protocol (EBGP) neighbor receives a local preference value, the EBGP neighbor ignores it.



Local Preference Case Study

What is the best path for router C to 65003, 65004, and 65005?

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-4

Example: Local Preference Case Study

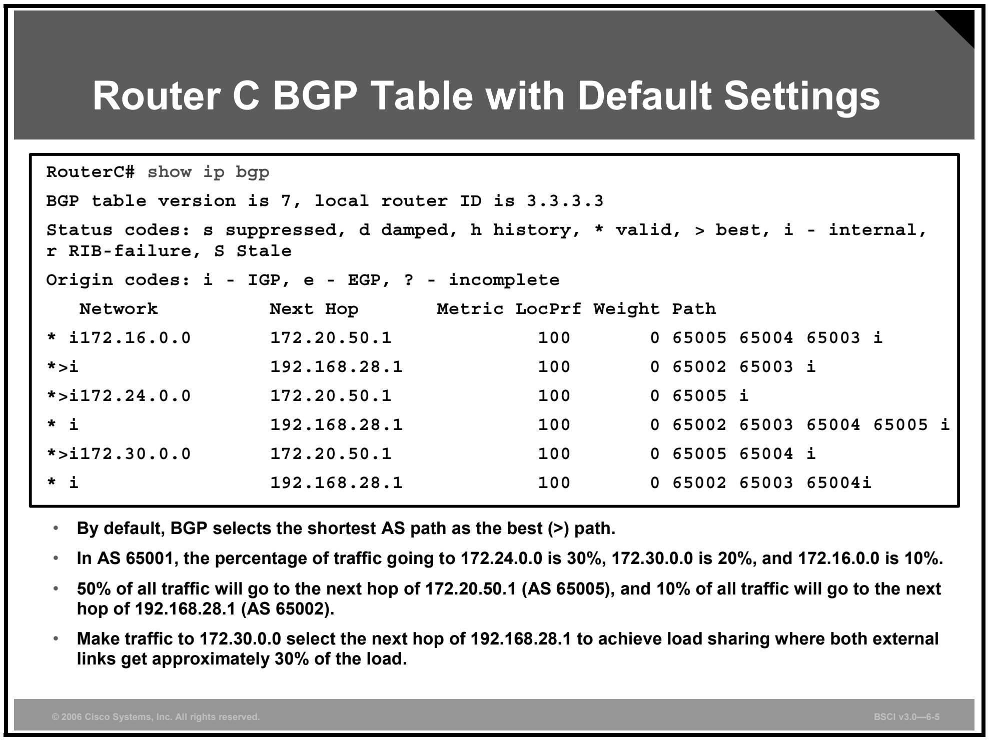
This figure illustrates an example network running BGP to demonstrate the manipulation of local preference using route maps in AS 65001.

From router C in AS 65001, the best path to network 172.16.0.0 in AS 65003 is determined in the following way:

- Steps 1 and 2 look at weight and local preference and use the default settings of weight equaling 0 and local preference equaling 100 for all routes that are learned from the IBGP neighbors of A and B.
- Step 3 does not help decide the best path because the three AS routes are not owned or originated by AS 65001.
- Step 4 prefers the shortest AS path; the options are two autonomous systems (65002, 65003) through router A or three autonomous systems through IBGP neighbor router B (65005, 65004, 65003). Thus, the shortest AS path from router C to AS 65003 is through router A.

The best path from router C to networks in AS 65005 is also selected by Step 4, the shortest AS path. The shortest path from router C to AS 65005 is through router B because it consists of one AS (65005) compared to four autonomous systems (65002, 65003, 65004, 65005) through router A.

The best path from router C to networks in AS 65004 is also selected by Step 4, the shortest AS path. The shortest path from router C to AS 65004 is through router B because it consists of two autonomous systems (65005, 65004) compared to three autonomous systems (65002, 65003, 65004) through router A.



Router C BGP Table with Default Settings

```
RouterC# show ip bgp
BGP table version is 7, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop       Metric LocPrf Weight Path
* i172.16.0.0       172.20.50.1             100       0 65005 65004 65003 i
*>i                 192.168.28.1            100       0 65002 65003 i
*>i172.24.0.0       172.20.50.1             100       0 65005 i
* i                 192.168.28.1            100       0 65002 65003 65004 65005 i
*>i172.30.0.0       172.20.50.1             100       0 65005 65004 i
* i                 192.168.28.1            100       0 65002 65003 65004i
```

- By default, BGP selects the shortest AS path as the best (>) path.
- In AS 65001, the percentage of traffic going to 172.24.0.0 is 30%, 172.30.0.0 is 20%, and 172.16.0.0 is 10%.
- 50% of all traffic will go to the next hop of 172.20.50.1 (AS 65005), and 10% of all traffic will go to the next hop of 192.168.28.1 (AS 65002).
- Make traffic to 172.30.0.0 select the next hop of 192.168.28.1 to achieve load sharing where both external links get approximately 30% of the load.

BSCI v3.0—6-5

Example: BGP Table with Default Settings

This figure demonstrates the BGP forwarding table on router C in AS 65001 with only default settings for BGP path selection. The diagram shows only the networks of interest to this example:

- 172.16.0.0 in AS 65003
- 172.24.0.0 in AS 65005
- 172.30.0.0 in AS 65004

The best path is indicated with “>” in the second column of the output.

Each network has two paths that are loop-free and synchronization-disabled and that have a valid next-hop address. All routes have a weight of 0 and a default local preference of 100; thus Steps 1 and 2 in the BGP path selection process are equal.

This router did not originate any of the routes (Step 3), so the process moves to Step 4, and BGP chooses the shortest AS path as follows:

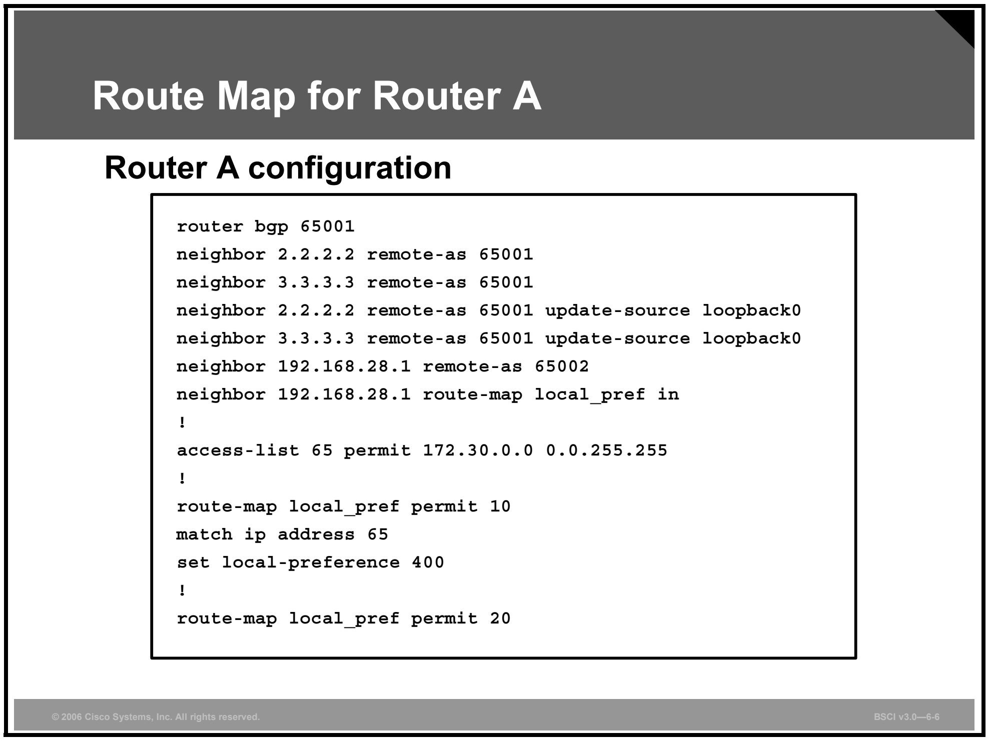
- For network 172.16.0.0, the shortest AS path of two autonomous systems (65002, 65003) is through the next hop of 192.168.28.1.
- For network 172.24.0.0, the shortest AS path of one AS (65005) is through the next hop of 172.20.50.1.
- For network 172.30.0.0, the shortest AS path of two autonomous systems (65005, 65004) is through the next hop of 172.20.50.1.

Neither router A nor router B is using the **next-hop-self** option in this example.

A traffic analysis reveals the following:

- The link going through router B to 172.20.50.1 is heavily used, and the link through router A to 192.168.28.1 is hardly used at all.
- The three largest-volume destination networks on the Internet from AS 65001 are 172.30.0.0, 172.24.0.0, and 172.16.0.0.
- 30 percent of all Internet traffic is going to network 172.24.0.0 (via router B); 20 percent is going to network 172.30.0.0 (via router B); and 10 percent is going to network 172.16.0.0 (via router A). The other 40 percent is going to other destinations. Only 10 percent of all traffic is using the link out of router A to 192.168.28.1, and 50 percent of all traffic is using the link out of router B to 172.20.50.1.

The network administrator has decided to divert traffic to network 172.30.0.0 and send it out router A to the next hop of 192.168.28.1, so that the loading between routers A and B is more balanced.



Route Map for Router A

Router A configuration

```
router bgp 65001
neighbor 2.2.2.2 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 2.2.2.2 remote-as 65001 update-source loopback0
neighbor 3.3.3.3 remote-as 65001 update-source loopback0
neighbor 192.168.28.1 remote-as 65002
neighbor 192.168.28.1 route-map local_pref in
!
access-list 65 permit 172.30.0.0 0.0.255.255
!
route-map local_pref permit 10
match ip address 65
set local-preference 400
!
route-map local_pref permit 20
```

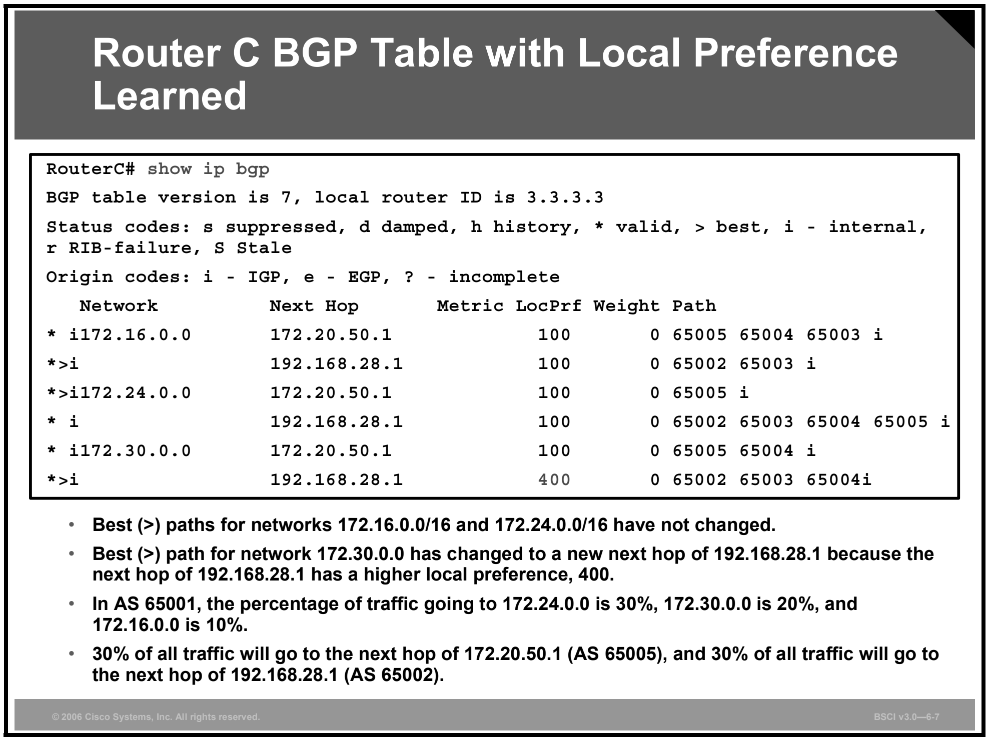
Example: Route Map for Router A

This figure demonstrates the use of a route map on router A to alter the network 172.30.0.0 BGP update from router X (192.168.28.1) to have a high local preference value of 400 so that it will be more preferred.

The first line of the route map is a permit statement with a sequence number of 10 for a route map called “local_pref”; this defines the first route-map statement. The match condition for that statement is checking all networks that are permitted by access list 65. Access list 65 permits all networks that start with the first two octets of 172.30.0.0; the route map sets those networks to a local preference of 400.

The second statement of the route map is a permit statement with a sequence number of 20 for the route map local_pref, but it does not have any match or set statements. This statement is a permit all statement for route maps. Because there are no match conditions for the remaining networks, they are all permitted with their current settings. In this case, the local preference for network 172.16.0.0 and 172.24.0.0 stays set at the default of 100. The sequence number of 20 is chosen for the second statement in case other policies at a later date have to be implemented before this permit all statement.

This route map is linked to neighbor 192.168.28.1 as an inbound route map; therefore, as router A receives updates from 192.168.28.1, it processes them through the local_pref route map and sets the local preference accordingly as the networks are placed into the router A BGP forwarding table.



Router C BGP Table with Local Preference Learned

```
RouterC# show ip bgp
BGP table version is 7, local router ID is 3.3.3.3
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop       Metric LocPrf Weight Path
* i172.16.0.0       172.20.50.1             100       0 65005 65004 65003 i
*>i                 192.168.28.1            100       0 65002 65003 i
*>i172.24.0.0       172.20.50.1             100       0 65005 i
* i                 192.168.28.1            100       0 65002 65003 65004 65005 i
* i172.30.0.0       172.20.50.1             100       0 65005 65004 i
*>i                 192.168.28.1            400       0 65002 65003 65004i
```

- **Best (>) paths for networks 172.16.0.0/16 and 172.24.0.0/16 have not changed.**
- **Best (>) path for network 172.30.0.0 has changed to a new next hop of 192.168.28.1 because the next hop of 192.168.28.1 has a higher local preference, 400.**
- **In AS 65001, the percentage of traffic going to 172.24.0.0 is 30%, 172.30.0.0 is 20%, and 172.16.0.0 is 10%.**
- **30% of all traffic will go to the next hop of 172.20.50.1 (AS 65005), and 30% of all traffic will go to the next hop of 192.168.28.1 (AS 65002).**

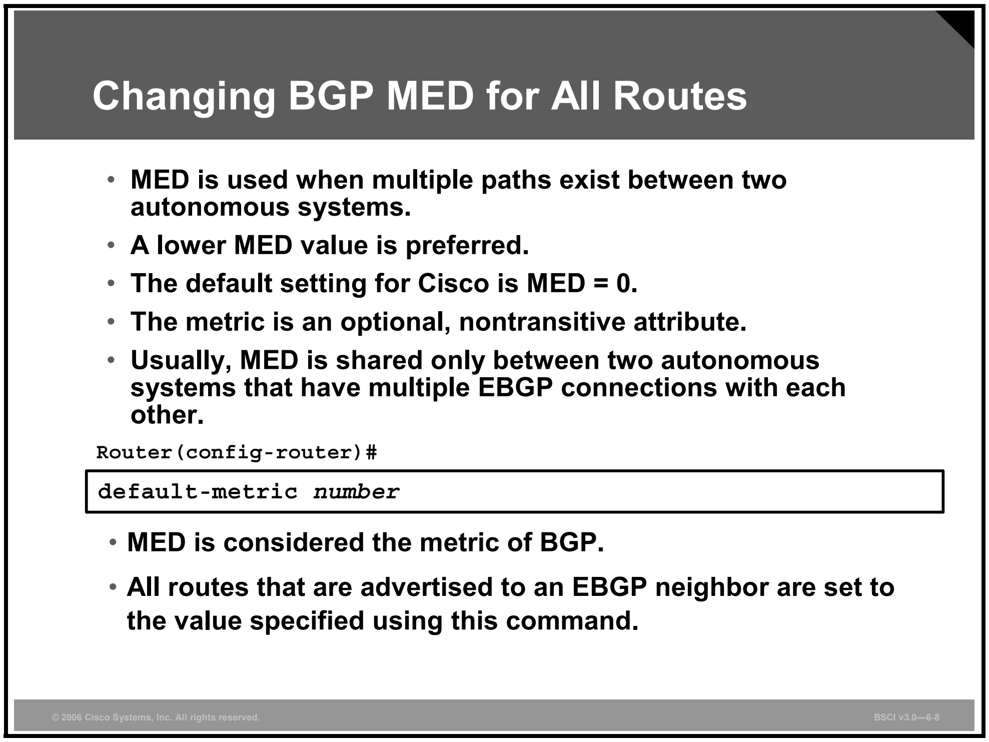
The figure shows the BGP forwarding table on router C in AS 65001 after the BGP session has been reset and illustrates that router C has learned about the new local preference value (400) coming from router A for network 172.30.0.0.

The only difference in this table compared to the previous example, which did not have local preference manipulation, is that the best route to network 172.30.0.0 is now through 192.168.28.1 because its local preference of 400 is higher than the local preference of 100 for the next hop of 172.20.50.1.

The AS path through 172.20.50.1 is still shorter than the path through 192.168.28.1, but AS path length is not evaluated until Step 4, while local preference is examined in Step 2. The higher local preference path was chosen as the best path.

Setting the MED with Route Maps

This topic describes how to use route maps to set the BGP MED attribute.



Changing BGP MED for All Routes

- **MED is used when multiple paths exist between two autonomous systems.**
- **A lower MED value is preferred.**
- **The default setting for Cisco is MED = 0.**
- **The metric is an optional, nontransitive attribute.**
- **Usually, MED is shared only between two autonomous systems that have multiple EBGP connections with each other.**

```
Router(config-router)#
default-metric number
```

- **MED is considered the metric of BGP.**
- **All routes that are advertised to an EBGP neighbor are set to the value specified using this command.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-8

Recall that the MED is used to decide how to enter an AS. It is used when multiple paths exist between two autonomous systems and one AS is trying to influence the incoming path from the other AS.

Because the MED is evaluated late in the BGP path selection process (Step 6), it usually has no influence on the BGP selection process. For example, an AS receiving an MED for a route can change its local preference to enable the AS to override what the other AS is advertising with its MED value.

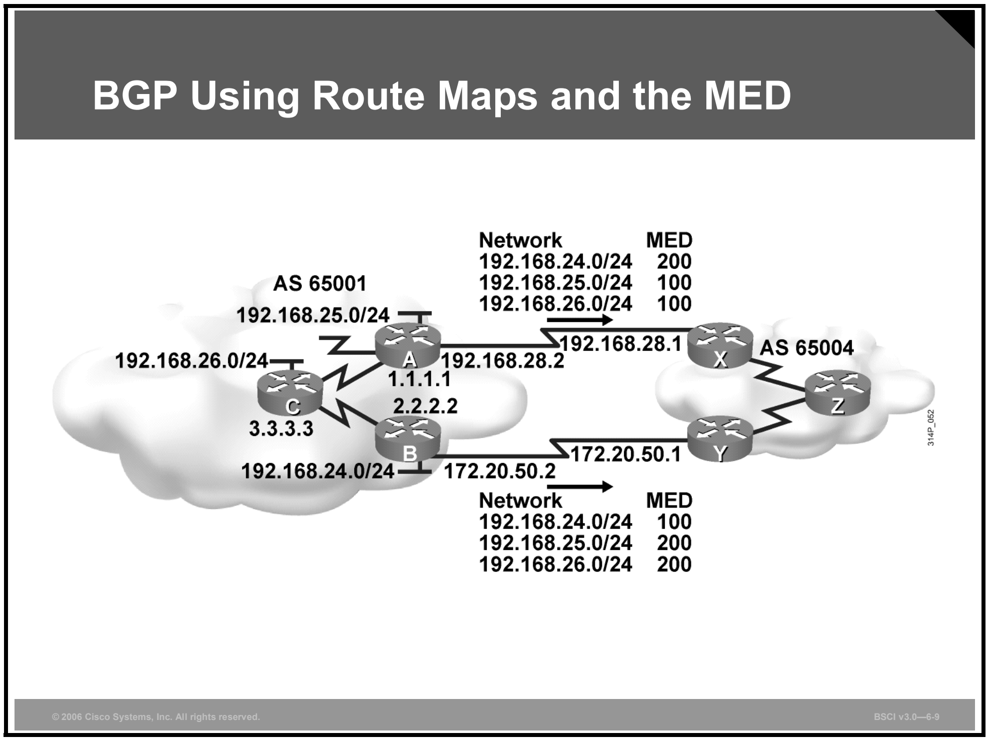
When BGP is comparing MED values for the same destination network in the path selection process, the lowest MED value is preferred.

The default MED value for each network that an AS owns and advertises to an EBGP neighbor is set to 0. To change this value, use the **default-metric** *number* command under the BGP process.

The default-metric Command Parameter

The table describes the syntax of the **default-metric** command.

Parameter	Description
number	(Optional) The value of the metric, which for BGP is the MED



BGP Using Route Maps and the MED

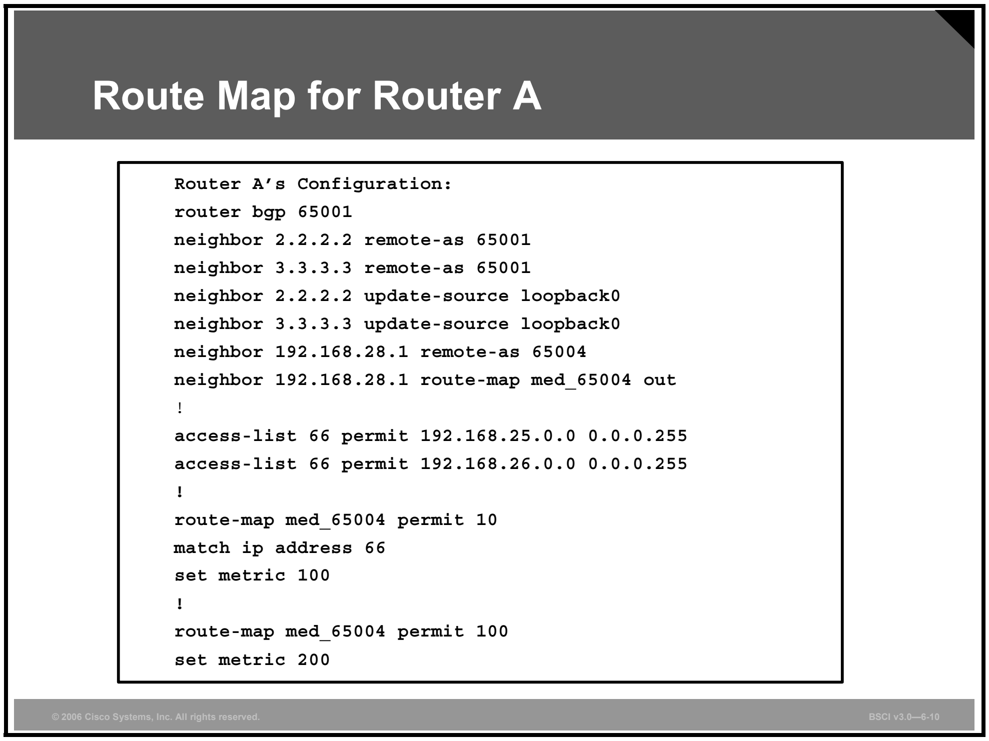
Network	MED
192.168.24.0/24	200
192.168.25.0/24	100
192.168.26.0/24	100

Network	MED
192.168.24.0/24	100
192.168.25.0/24	200
192.168.26.0/24	200

BSCI v3.0—6-9

Example: BGP Using Route Maps and the MED

This figure is used in the following configurations to demonstrate how to manipulate inbound traffic using route maps to change the BGP MED attribute. The intention of these route maps is to designate router A as the preferred path to reach networks 192.168.25.0/24 and 192.168.26.0/24 and to designate router B as the preferred path to reach network 192.168.24.0/24. The other networks should still be reachable through each router in case of a link or router failure.



Route Map for Router A

```
Router A's Configuration:
router bgp 65001
neighbor 2.2.2.2 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 2.2.2.2 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 192.168.28.1 remote-as 65004
neighbor 192.168.28.1 route-map med_65004 out
!
access-list 66 permit 192.168.25.0.0 0.0.0.255
access-list 66 permit 192.168.26.0.0 0.0.0.255
!
route-map med_65004 permit 10
match ip address 66
set metric 100
!
route-map med_65004 permit 100
set metric 200
```

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-10

The MED is set outbound when a router is advertising to an EBGP neighbor. In the configuration example for router A, a route map named "med_65004" is linked to neighbor 192.168.28.1 as an outbound route map.

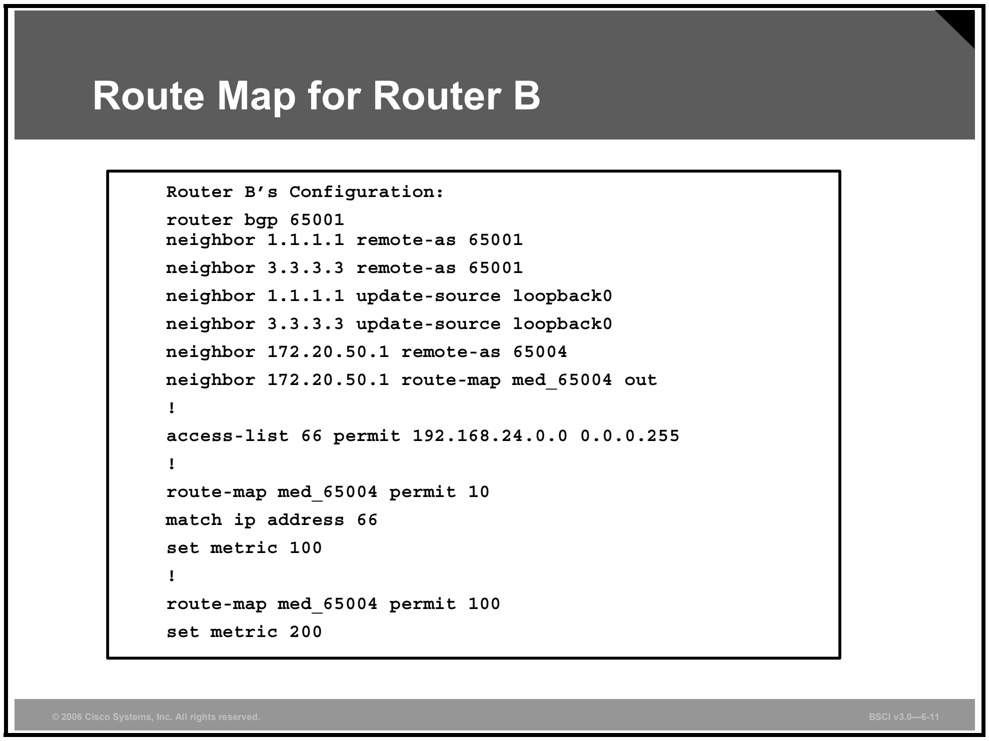
When router A sends an update to neighbor 192.168.28.1 (router X), it processes the outbound update through route map med_65004 and use a set statement to change any values that are specified, as long as the preceding match statement is met in that section of the route map.

The first line of the route map is a permit statement with a sequence number of 10 for the route map med_65004; this defines the first route-map statement. The match condition for this statement checks all networks that are permitted by access list 66. The first line of access list 66 permits any networks that start with the first three octets of 192.168.25.0, and the second line of access list 66 permits networks that start with the first three octets of 192.168.26.0.

All networks that are permitted by either of these lines are set to a MED of 100. All other networks are denied by this access list (there is an implicit deny all at the end of all access lists), so they are not set to a MED of 100; their MED is not changed. These other networks must proceed to the next route map statement in the med_65004 route map.

The second statement of the route map is a permit statement with a sequence number of 100 for the route map med_65004. The route map does not have any match statements, just a **set metric 200** command. This is a permit all statement for route maps.

Because the network administrator does not specify a match condition for this portion of the route map, all networks being processed through this section of the route map (sequence number 100) are permitted, and they are set to a MED of 200. If the network administrator did not set the MED to 200, by default it would have been a MED of 0. Because 0 is less than 100, the routes with a MED of 0 would have been the preferred paths to the networks in AS 65001.



Route Map for Router B

```
Router B's Configuration:
router bgp 65001
neighbor 1.1.1.1 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 1.1.1.1 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 172.20.50.1 remote-as 65004
neighbor 172.20.50.1 route-map med_65004 out
!
access-list 66 permit 192.168.24.0 0.0.0.255
!
route-map med_65004 permit 10
match ip address 66
set metric 100
!
route-map med_65004 permit 100
set metric 200
```

Similarly, in the configuration example for router B, a route map named "med_65004" is linked to neighbor 172.20.50.1 (router Y) as an outbound route map.

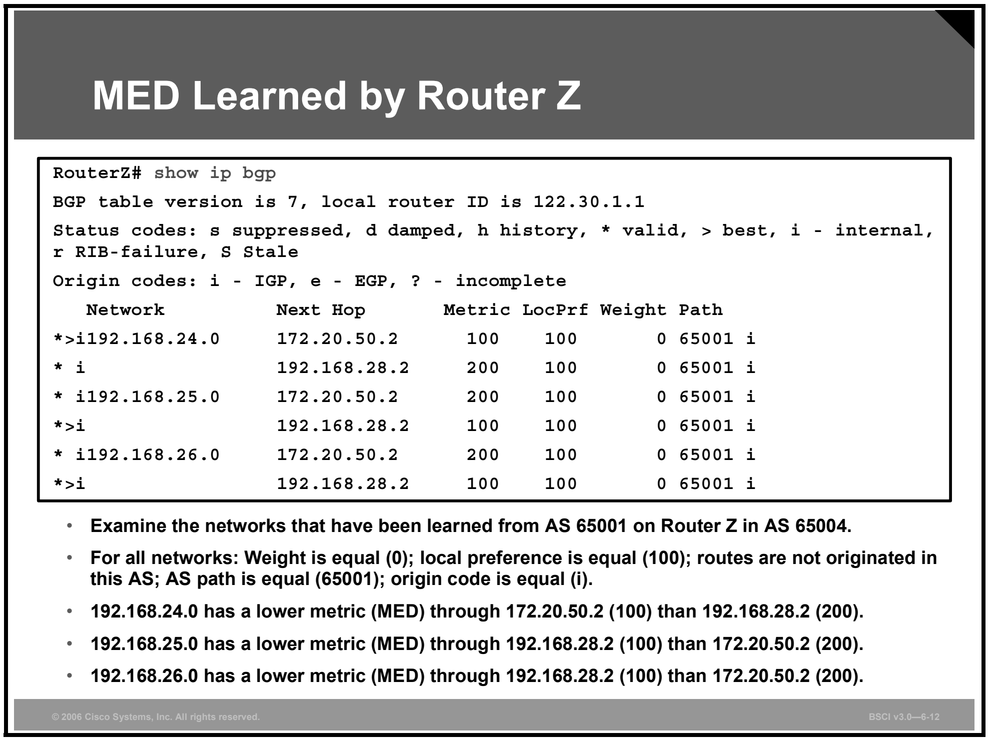
Before router B sends an update to neighbor 172.20.50.1, it will process the outbound update through route map med_65004 and use a set statement to change any values that are specified, as long as the preceding match statement is met in that section of the route map.

The first line of the route map is a permit statement with a sequence number of 10 for the route map med_65004, which defines the first route-map statement. The match condition for that line checks all networks that are permitted by access list 66. Access list 66 on router B permits any networks that start with the first three octets of 192.168.24.0.

Any networks that are permitted by this line are set to a MED of 100. All other networks are denied by this access list, so they are not set to a MED of 100. These other networks must proceed to the next route map statement in the med_65004 route map.

The second statement of the route map is a permit statement with a sequence number of 100 for the route map med_65004, but it does not have any match statements, just a **set metric 200** command. This is a permit all statement for route maps. Because the network administrator does not specify a match condition for this portion of the route map, all networks being processed through this topic are permitted, but they are set to a MED of 200.

If the network administrator did not set the MED to 200, by default it would have been set to a MED of 0. Because 0 is less than 100, the routes with a MED of 0 would have been the preferred paths to the networks in AS 65001.



MED Learned by Router Z

```
RouterZ# show ip bgp
BGP table version is 7, local router ID is 122.30.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop       Metric LocPrf Weight Path
*>i192.168.24.0     172.20.50.2      100    100       0 65001 i
* i                 192.168.28.2     200    100       0 65001 i
* i192.168.25.0     172.20.50.2      200    100       0 65001 i
*>i                 192.168.28.2     100    100       0 65001 i
* i192.168.26.0     172.20.50.2      200    100       0 65001 i
*>i                 192.168.28.2     100    100       0 65001 i
```

- **Examine the networks that have been learned from AS 65001 on Router Z in AS 65004.**
- **For all networks: Weight is equal (0); local preference is equal (100); routes are not originated in this AS; AS path is equal (65001); origin code is equal (i).**
- **192.168.24.0 has a lower metric (MED) through 172.20.50.2 (100) than 192.168.28.2 (200).**
- **192.168.25.0 has a lower metric (MED) through 192.168.28.2 (100) than 172.20.50.2 (200).**
- **192.168.26.0 has a lower metric (MED) through 192.168.28.2 (100) than 172.20.50.2 (200).**

The BGP forwarding table on router Z in AS 65004 displays the networks that have been learned from AS 65001. Other networks that do not affect this example have been omitted.

On router Z, there are multiple paths to reach each network. These paths all have valid next-hop addresses, have synchronization disabled, and are loop-free. All networks have a weight of 0 and a local preference of 100, so Steps 1 and 2 do not determine the best path.

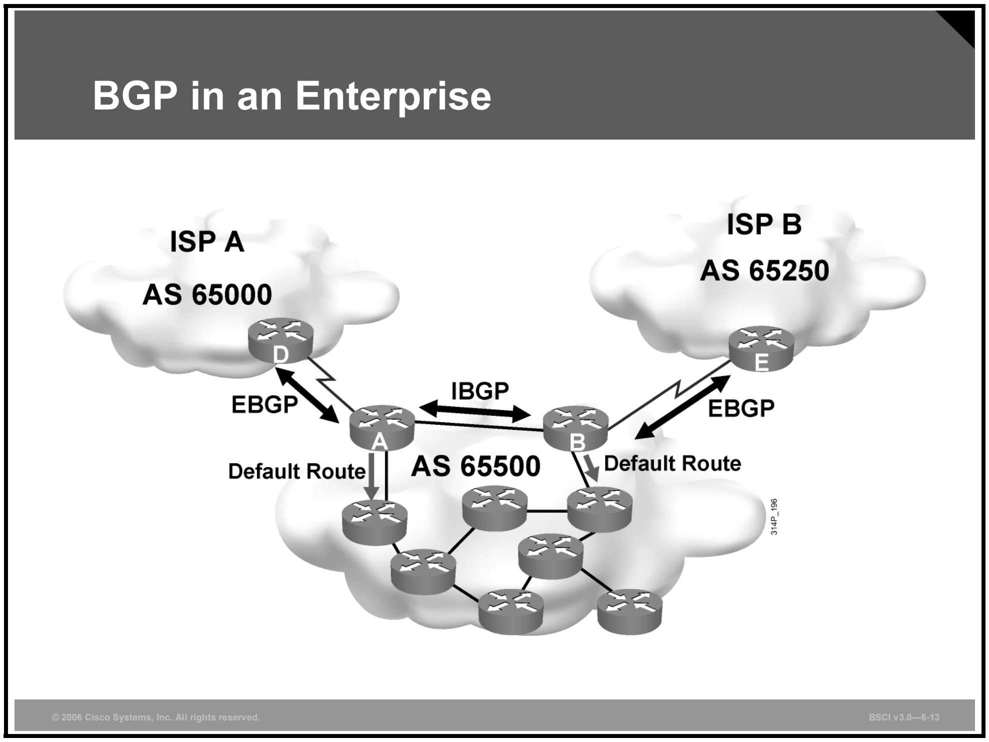
None of the routes were originated by this router or any router in AS 65004; all networks came from AS 65001, so Step 3 does not apply. All networks have an AS path of one AS (65001) and were introduced into BGP with network statements (“i” is the origin code), so Steps 4 and 5 are equal.

Step 6 states that BGP chooses the lowest MED if all preceding steps are equal or do not apply.

For network 192.168.24.0, the next hop of 172.20.50.2 has a lower MED than the next hop of 192.168.28.2; therefore, for network 192.168.24.0, the path through 172.20.50.2 is the preferred path. For networks 192.168.25.0 and 192.168.26.0, the next hop of 192.168.28.2 has a lower MED of 100 compared to the MED of 200 through the next hop of 172.20.50.2; therefore, 192.168.28.2 is the preferred path for those networks.

Implementing BGP in an Enterprise Network

This topic describes a typical enterprise BGP implementation.



BGP in an Enterprise

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-13

The figure depicts a typical enterprise BGP implementation. The enterprise is multihomed to two Internet service providers (ISPs), to increase the reliability and performance of its connection to the Internet.

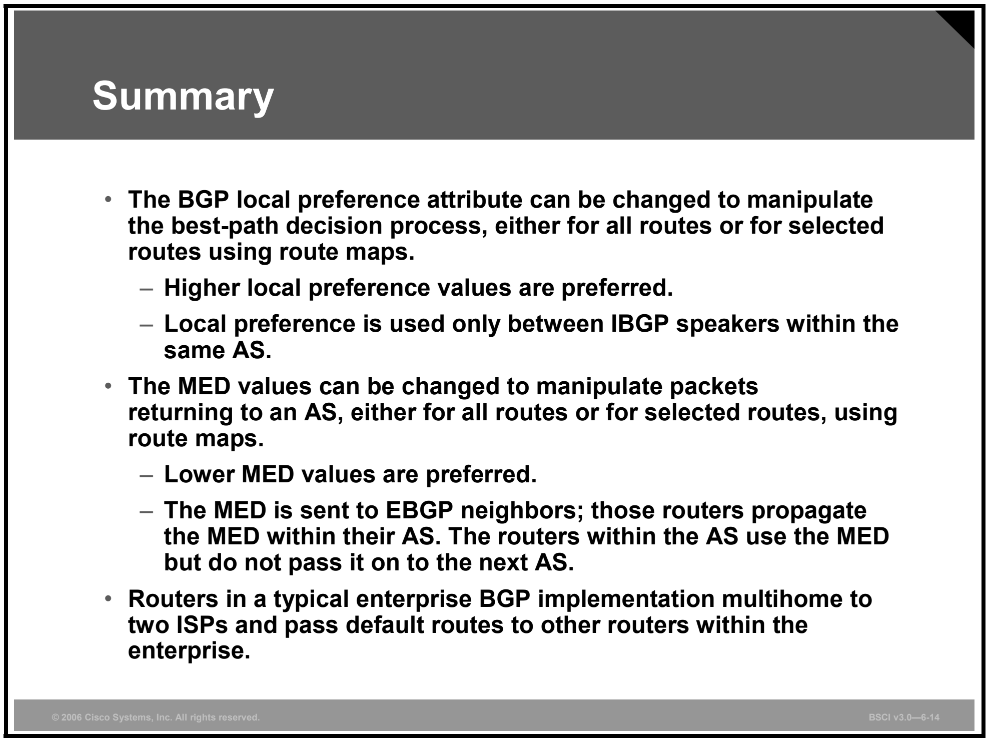
The ISPs may pass only default routes or may also pass other specific routes, or even all routes, to the enterprise.

The enterprise routers connected to the ISPs run EBGP with the ISP routers and IBGP between themselves; thus all routers in the transit path within the enterprise AS run IBGP. These routers pass default routes to the other routers in the enterprise rather than redistributing BGP into the interior routing protocol.

BGP attributes may be manipulated, using the methods discussed so far, by any of the routers running BGP to affect the path of the traffic to and from the autonomous systems.

Summary

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

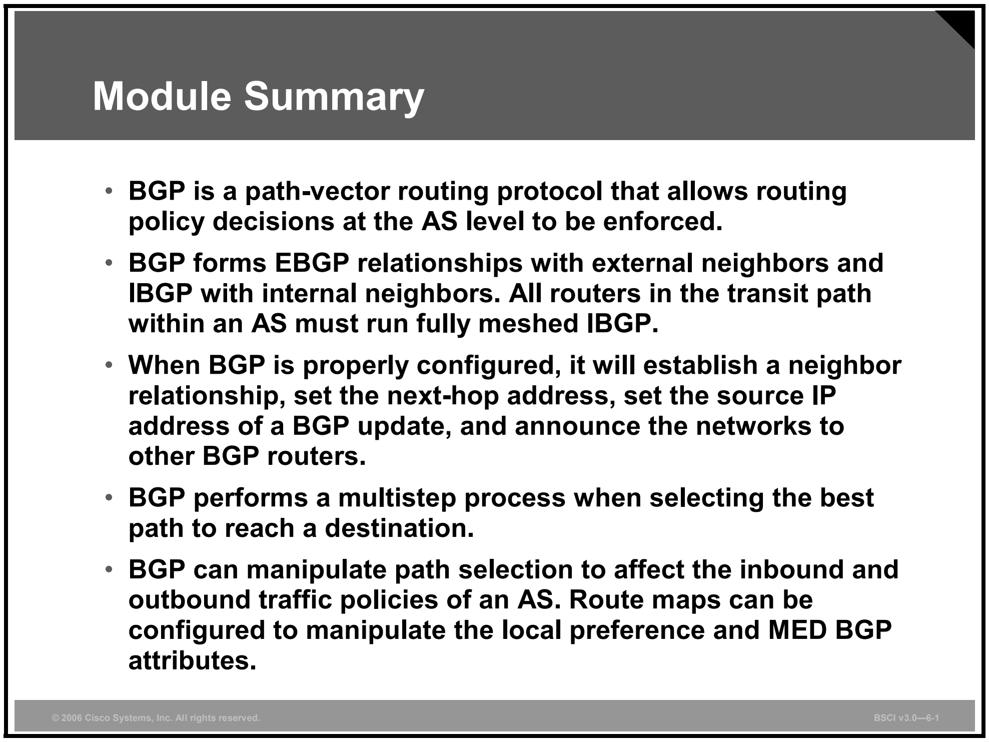


Summary

- **The BGP local preference attribute can be changed to manipulate the best-path decision process, either for all routes or for selected routes using route maps.**
  - **Higher local preference values are preferred.**
  - **Local preference is used only between IBGP speakers within the same AS.**
- **The MED values can be changed to manipulate packets returning to an AS, either for all routes or for selected routes, using route maps.**
  - **Lower MED values are preferred.**
  - **The MED is sent to EBGP neighbors; those routers propagate the MED within their AS. The routers within the AS use the MED but do not pass it on to the next AS.**
- **Routers in a typical enterprise BGP implementation multihome to two ISPs and pass default routes to other routers within the enterprise.**

Module Summary

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



Module Summary

- **BGP is a path-vector routing protocol that allows routing policy decisions at the AS level to be enforced.**
- **BGP forms EBGP relationships with external neighbors and IBGP with internal neighbors. All routers in the transit path within an AS must run fully meshed IBGP.**
- **When BGP is properly configured, it will establish a neighbor relationship, set the next-hop address, set the source IP address of a BGP update, and announce the networks to other BGP routers.**
- **BGP performs a multistep process when selecting the best path to reach a destination.**
- **BGP can manipulate path selection to affect the inbound and outbound traffic policies of an AS. Route maps can be configured to manipulate the local preference and MED BGP attributes.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—6-1

The Internet has proven to be a valuable tool to many companies, resulting in multiple redundant connections to many different Internet service providers (ISPs). The function of Border Gateway Protocol (BGP) is to provide alternatives to using default routes to control path selections.

References

For additional information, refer to these resources:

- RFCs 1772, 1773, 1774, 1930, 1966, 1997, 1998, 2042, 2385, 2439, 2545, 2547, 2796, 2858, 2918, 3065, 3107, 3392, 4223, and 4271.
- RFC 1518, *An Architecture for IP Address Allocation with CIDR.*
- RFC 1519, *Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy.*
- RFC 2050, *Internet Registry IP Allocation Guidelines.*

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) On what does BGP base the selection of the best path? (Source: Explaining BGP Concepts and Terminology)

A) speed
B) AS routing policy
C) number of routers to reach a destination network
D) bandwidth plus delay

Q2) Which routing method best describes BGP? (Source: Explaining BGP Concepts and Terminology)

A) distance vector
B) link-state
C) path-vector
D) hybrid of link-state and distance vector

Q3) Which two conditions are valid reasons to run BGP in an AS? (Choose two.) (Source: Explaining BGP Concepts and Terminology)

A) The AS is an ISP.
B) The AS has only a single connection to another AS.
C) Path and packet flow manipulation is required in this AS.
D) You have a limited understanding of BGP routing and route filtering.

Q4) Which BGP message establishes a BGP session and carries the hold time and the BGP router ID? (Source: Explaining BGP Concepts and Terminology)

A) BGP update message
B) BGP keepalive message
C) BGP open message
D) BGP notification message

Q5) Which two characteristics are true for BGP? (Choose two.) (Source: Explaining BGP Concepts and Terminology)

A) supports VLSM
B) supports CIDR
C) is an IGP
D) is not used for routing between autonomous systems

Q6) Which two statements are true for BGP route advertisements and path selection? (Choose two.) (Source: Explaining BGP Concepts and Terminology)

A) BGP selects the best path based on speed.
B) BGP routers exchange attributes.
C) BGP advertises paths.
D) BGP paths are not loop-free.

Q7) Which protocol does BGP use? (Source: Explaining BGP Concepts and Terminology)

A) UDP port 520
B) TCP port 179
C) IP protocol number 88
D) IP protocol number 89

Q8) Which component does a BGP update contain? (Source: Explaining BGP Concepts and Terminology)

A) multiple paths and multiple networks
B) a single path and multiple networks
C) a single path and a single network
D) multiple paths and a single network

Q9) Which BGP message is sent when an error condition is detected? (Source: Explaining BGP Concepts and Terminology)

A) BGP update message
B) BGP keepalive message
C) BGP open message
D) BGP notification message
E) BGP error message

Q10) What are three common ways to perform multihoming? (Choose three.) (Source: Explaining BGP Concepts and Terminology)

A) Each ISP passes only a default route to the AS.
B) Each ISP passes a default route and provider-owned specific routes to the AS.
C) Each ISP passes selected provider-owned routes but no default route to the AS.
D) Each ISP passes all routes to the AS.

Q11) Which two terms refer to routers that are configured to exchange BGP information with one another? (Choose two.) (Source: Explaining EBGP and IBGP)

A) BGP peer
B) BGP speaker
C) BGP router
D) BGP neighbor

Q12) By default, which two are conditions for routers to be EBGP neighbors? (Choose two.) (Source: Explaining EBGP and IBGP)

A) directly connected
B) in the same AS
C) in different autonomous systems
D) running an IGP between them to establish an adjacency

Q13) What are three ways to form an adjacency between IBGP neighbors by default? (Choose three.) (Source: Explaining EBGP and IBGP)

A) The neighbors can be directly connected.
B) The neighbors can be reachable from one another by static routes.
C) The neighbors can be reachable from one another by a dynamic internal routing protocol.
D) The neighbors can be in different AS.

Q14) Which statement about BGP is true? (Source: Explaining EBGP and IBGP)

A) Routes learned via IBGP are never sent to EBGP peers.
B) All the routers between IBGP neighbors must not be running BGP.
C) Routes learned via IBGP are never propagated to other IBGP peers.
D) Routes are never learned via IBGP.

Q15) Test your understanding of BGP terminology by matching terms with statements. Write the letter of the statement in front of the term that the statement describes. A statement can describe more than one term. Each term can match multiple statements, but choose only the statement that best describes the term. (Source: Explaining EBGP and IBGP)

Term

_____ 1. IBGP neighbors

_____ 2. BGP speakers

_____ 3. BGP neighbors

_____ 4. EBGP neighbors

_____ 5. BGP routers

_____ 6. BGP peers

Statement

A) These routers advertise BGP routing information.
B) This is a set of BGP routers that are explicitly configured to exchange BGP information. They have established a TCP connection with each other.
C) This is a set of BGP routers that, by default, can be multiple routers away from each other, but are in the same AS.
D) This is a set of BGP routers that, by default, must be directly connected and must be in different autonomous systems.

Q16) Which command indicates to a BGP router whether an IP address belongs to an IBGP or an EBGP neighbor? (Source: Configuring Basic BGP Operations)

A) **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
B) **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
C) **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system*
D) **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**

Q17) Which command sets the source IP address of a BGP update to be the IP address of a specific interface? (Source: Configuring Basic BGP Operations)

A) **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
B) **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
C) **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system*
D) **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**

Q18) Which one of these BGP network statements is valid? (Source: Configuring Basic BGP Operations)

A) **network 199.199.199.199 mask 255.255.255.0**
B) **network 191.200.100.0**
C) **network 172.16.1.0 mask 255.255.0.0**
D) **network 200.100.50.0**

Q19) Which state indicates that the router does not have a path to the neighbor IP address? (Source: Configuring Basic BGP Operations)

A) active
B) idle
C) established
D) open confirm

Q20) Which state indicates that an open message has been sent but a reply has not been received from the neighbor in more than 5 seconds? (Source: Configuring Basic BGP Operations)

A) active
B) idle
C) established
D) open confirm

Q21) Which command is the most disruptive method of resetting BGP sessions and should be avoided? (Source: Configuring Basic BGP Operations)

A) **clear ip bgp 192.168.200.1**
B) **clear ip bgp ***
C) **clear ip bgp 192.168.200.1 soft in**
D) **clear ip bgp 192.168.200.1 soft out**

Q22) Which command resends the routing table without resetting the TCP session and flags routes that the neighbor, 192.168.200.1, will not see anymore as “withdrawals”? (You should use this command if the outbound policy of a BGP router has changed.) (Source: Configuring Basic BGP Operations)

A) **clear ip bgp 192.168.200.1**
B) **clear ip bgp ***
C) **clear ip bgp 192.168.200.1 soft in**
D) **clear ip bgp 192.168.200.1 soft out**

Q23) Which command is used to administratively disable a BGP neighbor? (Source: Configuring Basic BGP Operations)

A) **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
B) **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
C) **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system*
D) **neighbor** {*ip-address* | *peer-group-name*} n**ext-hop-self**

Q24) Which command sets the next-hop address to be the source IP address of the update when advertising to a BGP neighbor? (Source: Configuring Basic BGP Operations)

A) **neighbor** {*ip-address* | *peer-group-name*} **shutdown**
B) **neighbor** {*ip-address* | *peer-group-name*} **update-source** *interface-type interface-number*
C) **neighbor** {*ip-address* | *peer-group-name*} **remote-as** *autonomous-system*
D) **neighbor** {*ip-address* | *peer-group-name*} **next-hop-self**

Q25) Which **clear ip bgp** command is the least intrusive for resetting a BGP session after changing outbound policy for neighbor 200.100.50.1? (Source: Configuring Basic BGP Operations)

A) **clear ip bgp ***
B) **clear ip bgp 200.100.50.1 soft out**
C) **clear ip bgp 200.100.50.1**
D) **clear ip bgp 200.100.50.1 soft in**

Q26) The **network** command that is used in the router BGP process identifies the interfaces out of which to advertise BGP updates. (Source: Configuring Basic BGP Operations)

A) true
B) false

Q27) A BGP router automatically peers with any other BGP router. (Source: Configuring Basic BGP Operations)

A) true
B) false

Q28) Which BGP neighbor state is the proper state for normal BGP neighbor operations? (Source: Configuring Basic BGP Operations)

A) active
B) open confirm
C) idle
D) established

Q29) Which command resets the TCP session only between a router and its neighbor, 192.168.200.1? (Source: Configuring Basic BGP Operations)

A) **clear ip bgp 192.168.200.1**
B) **clear ip bgp ***
C) **clear ip bgp 192.168.200.1 soft in**
D) **clear ip bgp 192.168.200.1 soft out**

Q30) In the output of the **show ip bgp** command, what does the “s” in front of the line for a network mean? (Source: Configuring Basic BGP Operations)

A) summarized network
B) subnet of a network
C) suppressed network
D) supernet of a network

Q31) When authenticating between two BGP routers, the same password must be configured on both routers. (Source: Configuring Basic BGP Operations)

A) true
B) false

Q32) Which description applies to the AS path attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive

Q33) Which description applies to the next-hop attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive

Q34) Which description applies to the origin attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive

Q35) Which description applies to the local preference attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive

Q36) Which description applies to the MED attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive

Q37) Which description applies to the weight attribute? (Source: Selecting a BGP Path)

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) proprietary to Cisco and not advertised to other BGP routers

Q38) BGP, by default, will load-balance across how many paths? (Source: Selecting a BGP Path)

A) 1
B) 2
C) 4
D) 6

Q39) Which path will BGP prefer when using the weight attribute? (Source: Selecting a BGP Path)

A) higher weight
B) lower weight

Q40) Which path will BGP prefer when using the local preference attribute? (Source: Selecting a BGP Path)

A) higher local preference
B) lower local preference

Q41) Which path will BGP prefer when using the MED attribute? (Source: Selecting a BGP Path)

A) higher MED
B) lower MED

Q42) Match the BGP attributes to their characteristics by writing the letter of the characteristic that best describes the attribute in front of the attribute. The same characteristic can be used for more than one attribute, but each attribute will have only one characteristic. (Source: Selecting a BGP Path)

Attribute

_____ 1. AS path

_____ 2. next-hop

_____ 3. local preference

_____ 4. MED

_____ 5. origin

_____ 6. weight

_____ 7. atomic aggregate

Characteristic

A) well-known mandatory
B) well-known discretionary
C) optional transitive
D) optional nontransitive
E) proprietary to Cisco

Q43) Place the BGP selection criteria in order from the first step to the last step evaluated to select the BGP path that is submitted to the IP routing table. (Source: Selecting a BGP Path)

A) _____ prefer the path with the lowest neighbor BGP router ID
B) _____ prefer the lowest MED
C) _____ prefer the shortest AS path
D) _____ prefer the oldest route for EBGP paths
E) _____ prefer the lowest origin code (IGP < EGP < incomplete)
F) _____ prefer the highest weight
G) _____ prefer the path through the closest IGP neighbor
H) _____ prefer the highest local preference
I) _____ prefer the route that was originated by the local router
J) _____ prefer an EBGP path over IBGP path
K) _____ prefer the lowest neighbor IP address

Q44) Which two statements are true regarding local preference? (Choose two.) (Source: Using Route Maps to Manipulate Basic BGP Paths)

A) The higher value for local preference is preferred.
B) Local preference is used only between EBGP neighbors.
C) The lower value for local preference is preferred.
D) Local preference is used only between IBGP neighbors.

Q45) Which two statements are true regarding the MED? (Choose two.) (Source: Using Route Maps to Manipulate Basic BGP Paths)

A) The higher value for the MED is preferred.
B) The MED is exchanged between autonomous systems.
C) The lower value for the MED is preferred.
D) The MED is local to an AS.

Q46) Which command changes the MED for all routes? (Source: Using Route Maps to Manipulate Basic BGP Paths)

A) **bgp med** *number*
B) **default-metric** *number*
C) **default-med** *number*
D) **set med** *number*
E) **bgp default-metric** *number*

Q47) Which command is used within a route map to change the local preference value? (Source: Using Route Maps to Manipulate Basic BGP Paths)

A) **bgp default local-preference** *value*
B) **default local-preference** *value*
C) **set local-preference** *value*
D) **set metric** *value*

Module Self-Check Answer Key

Q1)	B
Q2)	C
Q3)	A, C
Q4)	C
Q5)	A, B
Q6)	B, C
Q7)	B
Q8)	B
Q9)	D
Q10)	A, B, D
Q11)	A, D
Q12)	A, C
Q13)	A, B, C
Q14)	C
Q15)	1 = C, 2 = A, 3 = B, 4 = D, 5 = A, 6 = B
Q16)	C
Q17)	B
Q18)	D
Q19)	B
Q20)	A
Q21)	B
Q22)	D
Q23)	A
Q24)	D
Q25)	B
Q26)	B
Q27)	B
Q28)	D
Q29)	A
Q30)	C
Q31)	A
Q32)	A
Q33)	A
Q34)	A
Q35)	B

Q36)	D
Q37)	D
Q38)	A
Q39)	A
Q40)	A
Q41)	B
Q42)	1 = A, 2 = A, 3 = B, 4 = D, 5 = A, 6 = E, 7 = B
Q43)	A = 10, B = 6, C = 4, D = 9, E = 5, F = 1, G = 8, H = 2, I = 3, J = 7, K = 11
Q44)	A, D
Q45)	B, C
Q46)	B
Q47)	C

Module 7

Implementing Multicast

Overview

This module provides a moderately detailed overview of IP multicast. It explains the applications that utilize multicast technology and the benefit that multicast provides to the user of the applications.

IP multicast includes an addressing standard, methodologies for multicast users to become members of groups (Internet Group Management Protocol [IGMP]), source and shared trees, and a multicast routing protocol (Protocol Independent Multicast [PIM]). Cisco IOS command-line interface (CLI) configurations are included for implementation of IP multicast on Cisco Systems devices.

Module Objectives

Upon completing this module, you will be able to implement and verify multicast forwarding using PIM and related protocols. This ability includes being able to meet these objectives:

- Define multicast, as well as the concepts and the network components that are required to make multicast work
- Define and implement IGMP and resolve frame forwarding issues in Ethernet switching
- Describe and select multicast routing protocols
- Configure, verify, and test PIM sparse-dense mode, and verify IGMP snooping

Lesson 1

Explaining Multicast

Overview

This lesson introduces the concept of a multicast group, the types of applications that benefit from the use of multicasting, and why these benefits occur. Basic IP multicast addressing is also covered.

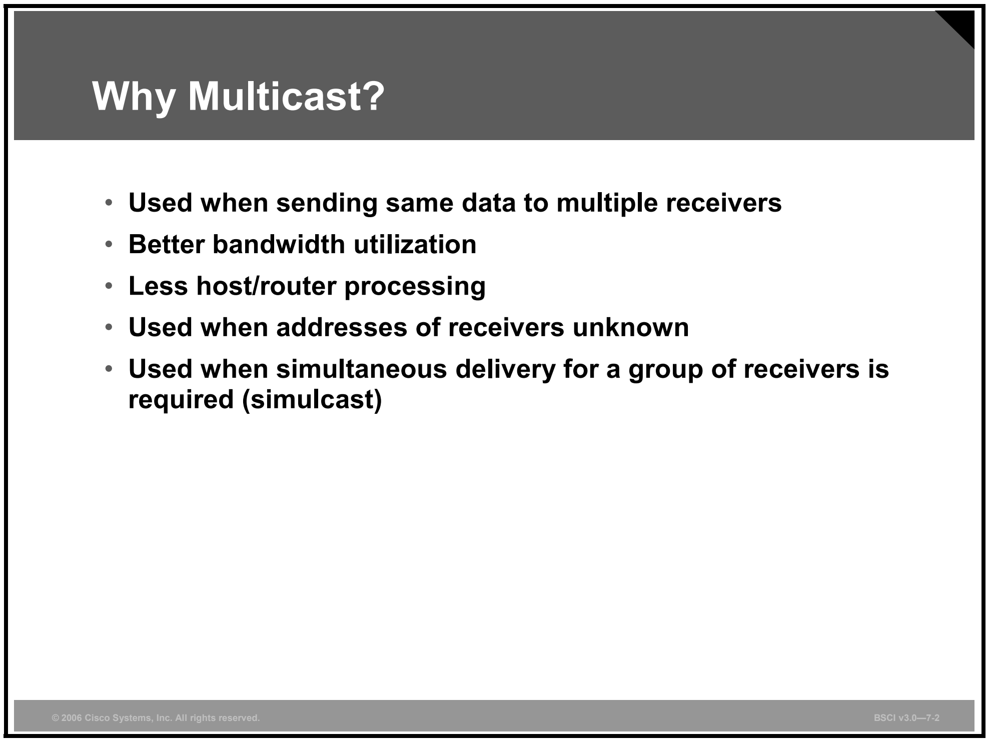
Objectives

Upon completing this lesson, you will be able to define multicast, as well as the concepts and the network components that are required to make multicast work. This ability includes being able to meet these objectives:

- Explain the concept of an IP multicast group
- Describe the types of multicast addresses

Explaining the Multicast Group

This topic describes the concept and use of an IP multicast group.



Why Multicast?

- Used when sending same data to multiple receivers
- Better bandwidth utilization
- Less host/router processing
- Used when addresses of receivers unknown
- Used when simultaneous delivery for a group of receivers is required (simulcast)

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—7-2

Multicast may be used to send the same data packets to multiple receivers.

By sending the data packets to multiple receivers, the packets are not duplicated for every receiver but are sent in a single stream, where downstream routers perform packet multiplication over receiving links.

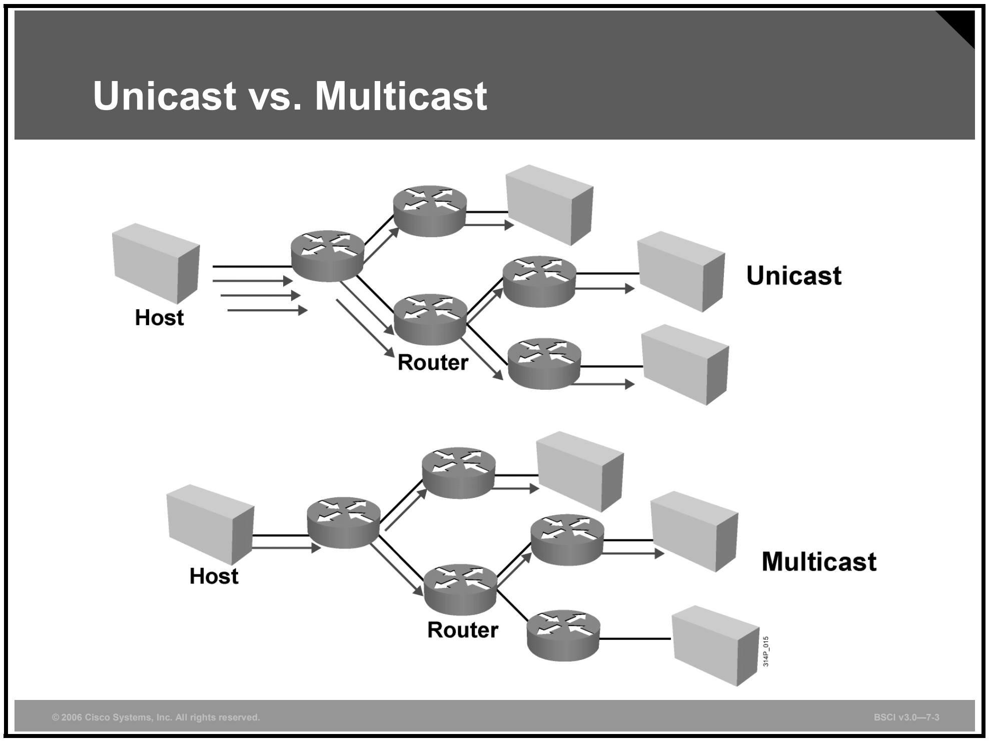
Routers process fewer packets because they receive only a single copy of the packet.

Because downstream routers perform packet multiplication and delivery to receivers, the sender, or source of multicast traffic, does not have to know the unicast addresses of the receiver.

Simulcast—simultaneous delivery for a group of receivers—may be used for several purposes, including audio or video streaming, news and similar data delivery, and deploying software upgrades.

To send data to multiple destinations using unicast, the sender has to send the same data flow to each receiver separately The sender has to make copies of the same packet and send them once for each receiver.

Some web technologies (for example, webcasting) use a “push” method to deliver the same data to multiple users. Instead of users clicking a link to get the data, the data is delivered automatically. Users first have to subscribe to a channel to receive the data, and after that, the data is periodically pushed to the user. The problem with the webcast is that the transport is still done using unicast.



Unicast vs. Multicast

BSCI v3.0—7-3

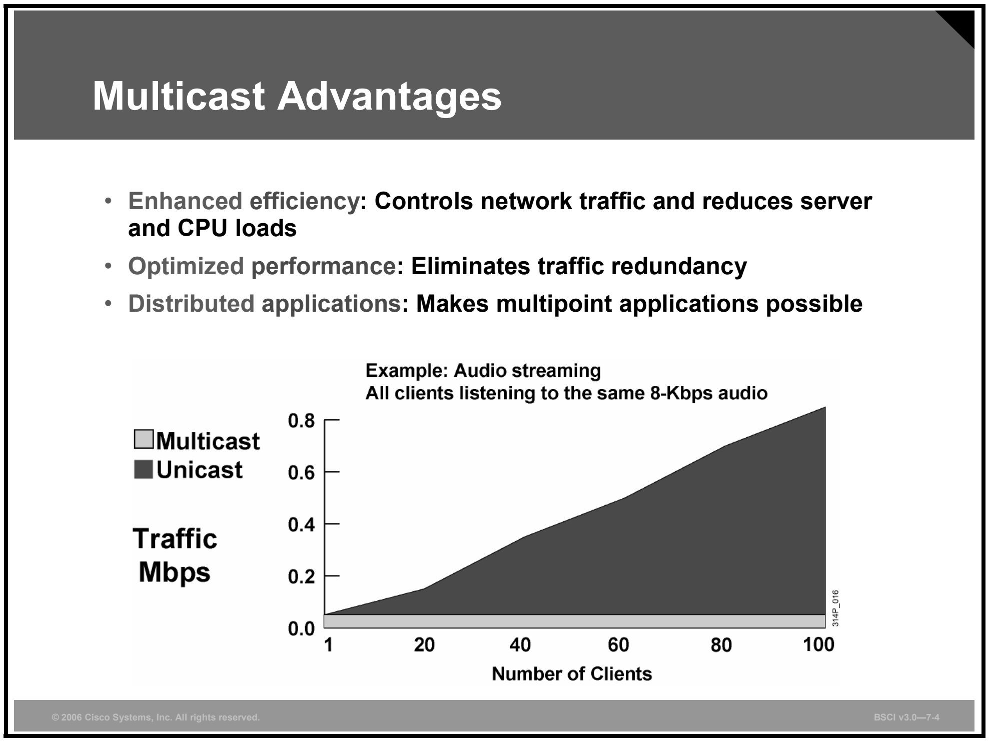
Unicast transmission sends multiple copies of data, one copy for each receiver.

The unicast example in the figure shows a host transmitting three copies of data and a network forwarding each packet to three separate receivers. The host may send to only one receiver at a time, because it has to create a different packet destination address for each receiver.

Multicast transmission sends a single copy of data to multiple receivers.

The multicast example shows a host transmitting one copy of the data and a network replicating the packet at the last possible hop for each receiver. Each packet exists only in a single copy on any given network. The host may send to multiple receivers simultaneously because it is sending only one packet.

Downstream multicast routers replicate and forward the data packet to all of those branches where there may be receivers.



Multicast Advantages

- Enhanced efficiency: Controls network traffic and reduces server and CPU loads
- Optimized performance: Eliminates traffic redundancy
- Distributed applications: Makes multipoint applications possible

BSCI v3.0—7-4

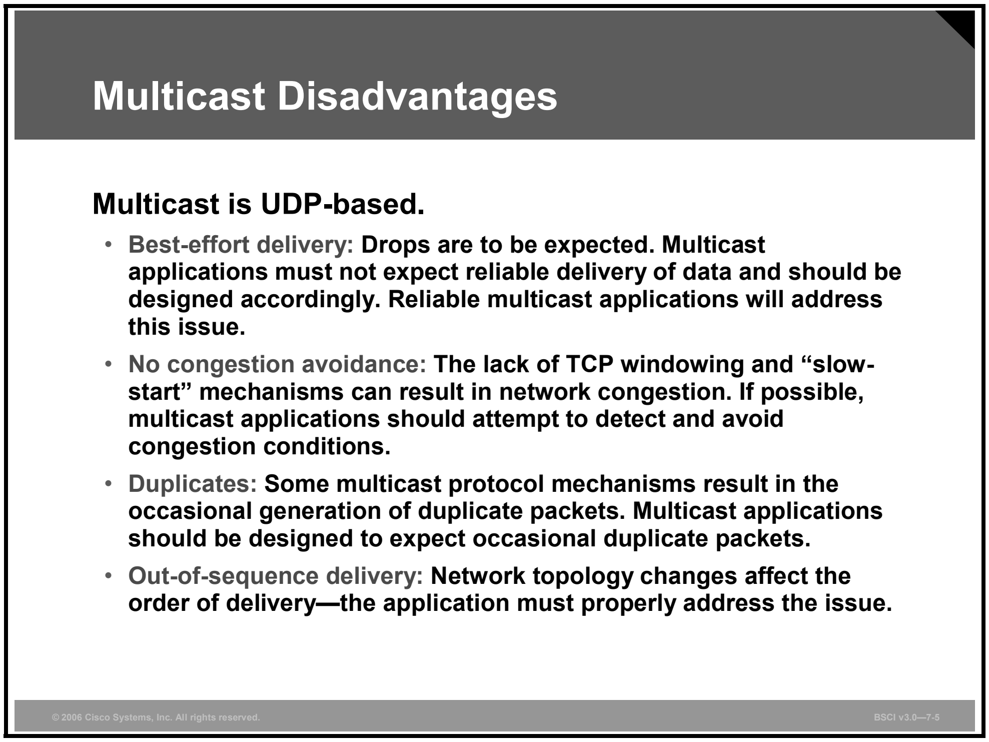
Multicast transmission provides many advantages over unicast transmission in a one-to-many or many-to-many environment:

- **Enhanced efficiency:** Available network bandwidth is utilized more efficiently because multiple streams of data are replaced with a single transmission.
- **Optimized performance:** Fewer copies of the data require forwarding and processing.
- **Distributed applications:** Multipoint applications will not be possible with unicast as demand and usage grows, because unicast transmission will not scale (traffic level and clients increase at a 1:1 rate with unicast transmission).

There are other multicast advantages:

- For the equivalent amount of multicast traffic, the sender needs much less processing power and bandwidth.
- Multicast packets do not impose as high a rate of bandwidth utilization as unicast packets, so there is a greater possibility that they will arrive almost simultaneously at the receivers.

Multicast enables a whole range of new applications that were not possible on unicast (for example, video on demand [VoD]).



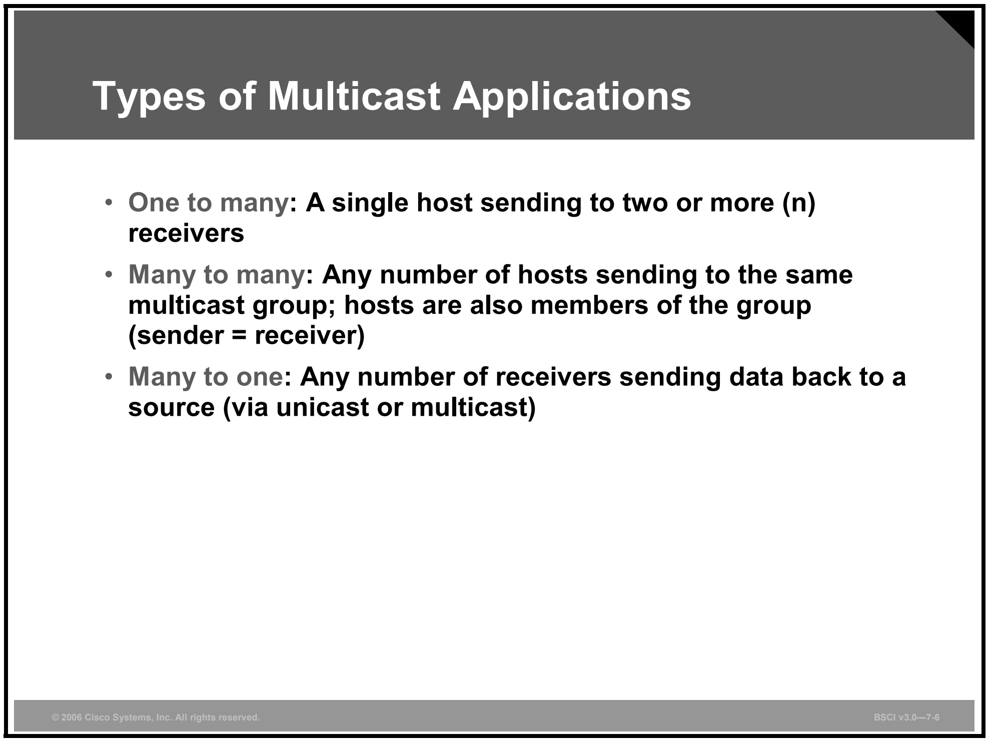
Multicast Disadvantages

Multicast is UDP-based.

- **Best-effort delivery: Drops are to be expected. Multicast applications must not expect reliable delivery of data and should be designed accordingly. Reliable multicast applications will address this issue.**
- **No congestion avoidance: The lack of TCP windowing and "slow-start" mechanisms can result in network congestion. If possible, multicast applications should attempt to detect and avoid congestion conditions.**
- **Duplicates: Some multicast protocol mechanisms result in the occasional generation of duplicate packets. Multicast applications should be designed to expect occasional duplicate packets.**
- **Out-of-sequence delivery: Network topology changes affect the order of delivery—the application must properly address the issue.**

There are also some disadvantages of multicast that need to be considered:

- Most multicast applications are User Datagram Protocol (UDP)-based. This foundation results in some undesirable consequences compared to similar unicast TCP applications.
- Best-effort delivery results in occasional packet drops. Many multicast applications that operate in real time (for example, video and audio) may be affected by these losses. Also, requesting retransmission of the lost data at the application layer in these not-quite-real-time applications is not feasible.
  - Heavy drops on voice applications result in jerky, missed speech patterns that can make the content unintelligible when the drop rate gets too high.
  - Moderate to heavy drops in video are sometimes better tolerated by the human eye and appear as unusual "artifacts" in the picture. However, some compression algorithms may be severely affected by low drop rates, which will cause the picture to become jerky or to freeze for several seconds while the decompression algorithm recovers.
- Lack of congestion control may result in overall network degradation as the popularity of UDP-based multicast applications grow.
- Duplicate packets may occasionally be generated as multicast network topologies change. Applications must expect occasional duplicate packets to arrive and must be designed accordingly.
- Out-of-sequence delivery of packets to the application may also result during network topology changes or other network events that affect the flow of multicast traffic.
- UDP has no reliability mechanisms, so reliability issues have to be addressed in multicast applications where reliable data transfer is necessary.
- The issue of restricting multicast traffic to only a selected group of receivers, in other words, eavesdropping issues, has not yet been sufficiently resolved.
- Some commercial applications become possible only when reliability and security issues are fully resolved (for example, financial data delivery).



Types of Multicast Applications

- One to many: **A single host sending to two or more (n) receivers**
- Many to many: **Any number of hosts sending to the same multicast group; hosts are also members of the group (sender = receiver)**
- Many to one: **Any number of receivers sending data back to a source (via unicast or multicast)**

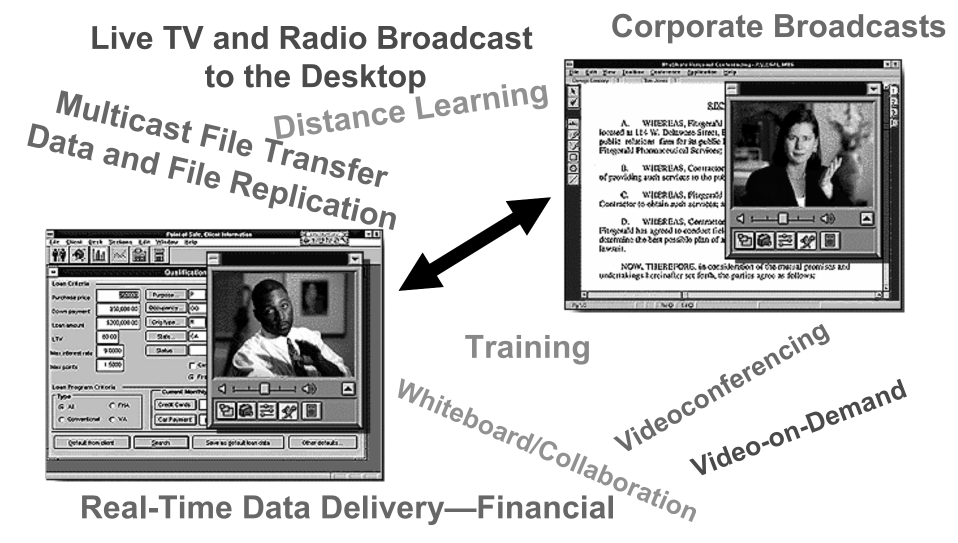
BSCI v3.0—7-6

There are various types of multicast applications. Here are two of the most common models:

- One-to-many model, where one sender sends data to many receivers
  - This type of application may be used for audio or video distribution, push media, announcements, monitoring, and so on.
  - If a one-to-many application needs feedback from receivers, it becomes a many-to-many application.
- Many-to-many model, where a host can be a sender and a receiver simultaneously or where two or more receivers also act as senders.
  - Receiving data from several sources increases the complexity of applications and creates different management challenges.
  - Using a many-to-many multicast concept as a foundation, a whole new range of applications may be built (for example, collaboration, concurrent processing, and distributed interactive simulations).

Other models (for example, many-to-one, where many receivers are sending data back to one sender, or few-to-many) are also used, especially in financial applications and networks. Many-to-one multicasts may be used for resource discovery, data collection, auctions, polling, and similar applications, but building them imposes several challenges.

IP Multicast Applications



Live TV and Radio Broadcast to the Desktop

Corporate Broadcasts

Multicast File Transfer

Distance Learning

Data and File Replication

Training

Whiteboard/Collaboration

Videoconferencing

Video-on-Demand

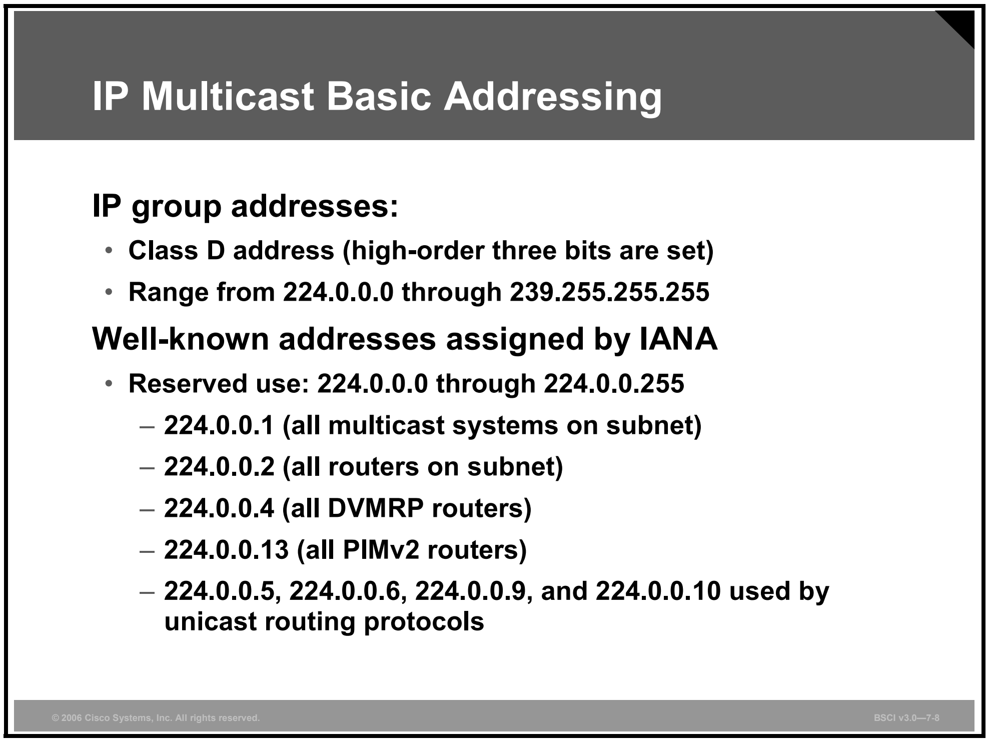
Real-Time Data Delivery—Financial

Many new multicast applications are emerging as demand for them grows.

- Real-time applications include live broadcasts, financial data delivery, whiteboard collaboration, and videoconferencing.
- Nonreal-time applications include file transfer, data and file replication, and VoD. Ghosting multiple PC images simultaneously is a common file transfer application.

IP Multicast Addresses

This topic describes the types of IP multicast addresses.



IP Multicast Basic Addressing

IP group addresses:

- **Class D address (high-order three bits are set)**
- **Range from 224.0.0.0 through 239.255.255.255**

Well-known addresses assigned by IANA

- **Reserved use: 224.0.0.0 through 224.0.0.255**
  - **224.0.0.1 (all multicast systems on subnet)**
  - **224.0.0.2 (all routers on subnet)**
  - **224.0.0.4 (all DVMRP routers)**
  - **224.0.0.13 (all PIMv2 routers)**
  - **224.0.0.5, 224.0.0.6, 224.0.0.9, and 224.0.0.10 used by unicast routing protocols**

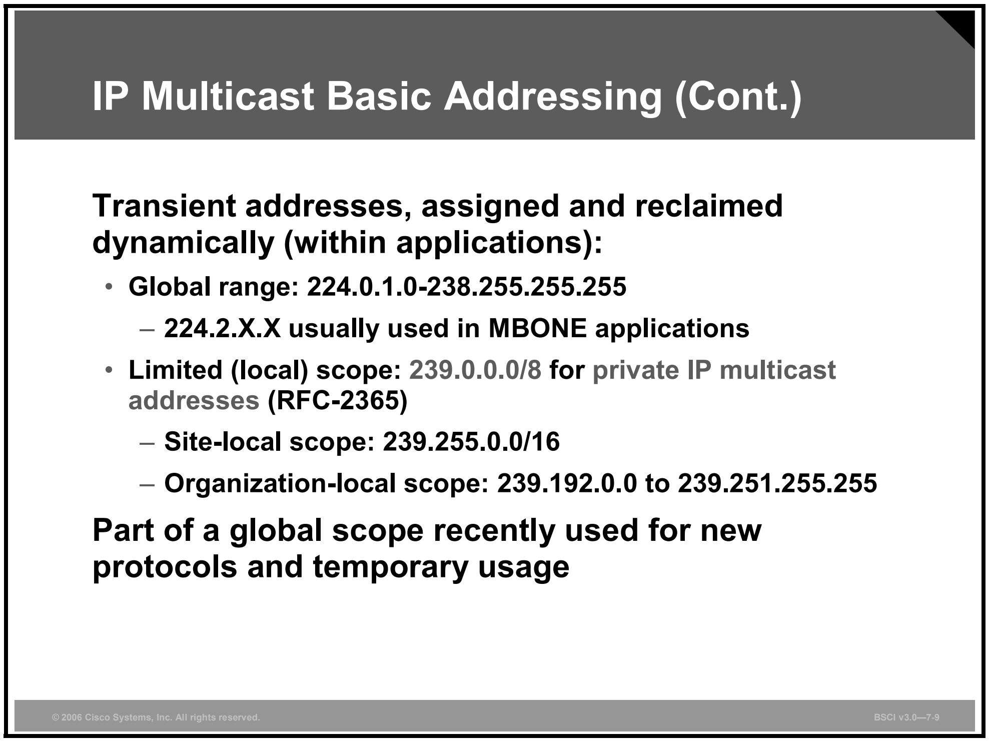
BSCI v3.0—7-8

Multicast IP addresses use the Class D address space and are denoted by the high-order three bits set to 1s (1110). Therefore, the multicast IP address range is from 224.0.0.0 through 239.255.255.255.

The multicast IP address space is separated into the following address groups:

- Local scope addresses are addresses 224.0.0.0 through 224.0.0.255 and are reserved by Internet Assigned Numbers Authority (IANA) for network protocol use. Multicasts in this range are never forwarded off the local network, regardless of Time to Live (TTL), and usually the TTL is set to 1. Here are examples of local multicast addresses:

—	224.0.0.1	All hosts
—	224.0.0.2	All multicast routers
—	224.0.0.4	All Distance Vector Multicast Routing Protocol (DVMRP) routers
—	224.0.0.5	All Open Shortest Path First Protocol (OSPF) routers
—	224.0.0.6	All OSPF designated routers (DRs)
—	224.0.0.9	All Routing Information Protocol version 2 (RIPv2) routers
—	224.0.0.10	All Enhanced Interior Gateway Routing Protocol (EIGRP) routers



IP Multicast Basic Addressing (Cont.)

Transient addresses, assigned and reclaimed dynamically (within applications):

- **Global range: 224.0.1.0-238.255.255.255**
  - **224.2.X.X usually used in MBONE applications**
- **Limited (local) scope: 239.0.0.0/8 for private IP multicast addresses (RFC-2365)**
  - **Site-local scope: 239.255.0.0/16**
  - **Organization-local scope: 239.192.0.0 to 239.251.255.255**

Part of a global scope recently used for new protocols and temporary usage

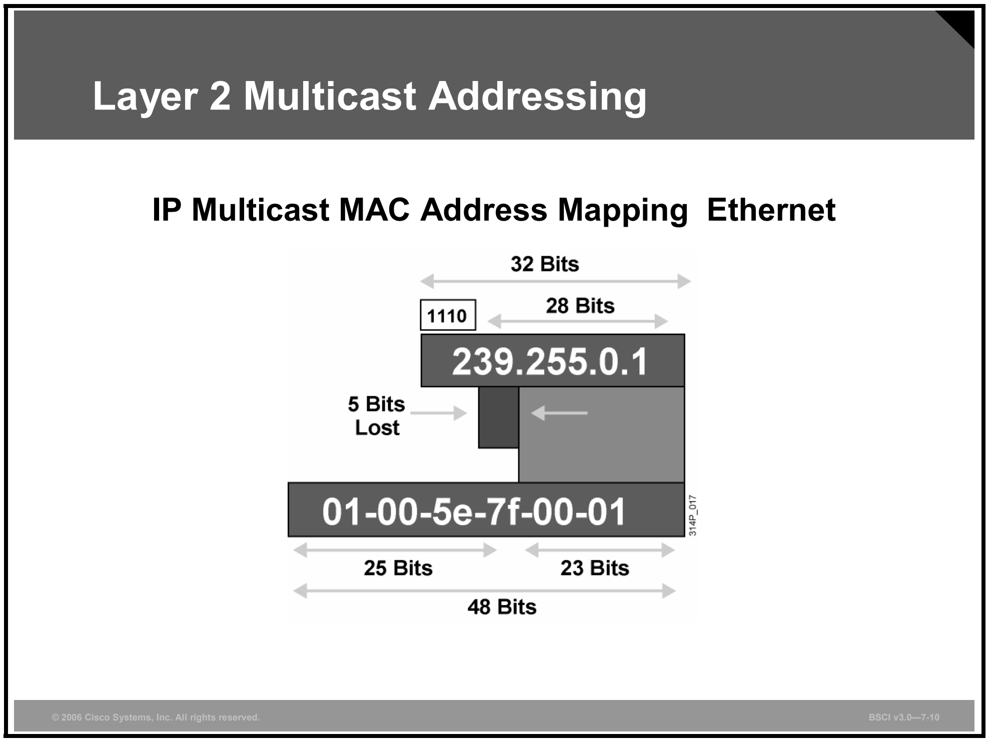
© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—7-9

For Multicast applications, transient addresses are dynamically assigned and then returned for others to use when no longer needed.. Two address types are:

- Global scope addresses are addresses 224.0.1.0 through 238.255.255.255, and are allocated dynamically throughout the Internet. For example, the 224.2.X.X range is used in Mbone applications. Mbone stands for Multicast Backbone and is a collection of Internet routers that support IP multicasting. The Mbone is used as a virtual network (multicast channel) on which various public and private audio and video programs are sent. Mbone was originated by the Internet Engineering Task Force (IETF) in an effort to multicast audio and video meetings.
- Administratively scoped addresses are addresses 239.0.0.0 through 239.255.255.255, and they are reserved for use inside private domains.

The administratively scoped multicast address space is divided into the following scopes per IANA:

- Site-local scope (239.255.0.0/16, with 239.252.0.0/16, 239.253.0.0/16, and 239.254.0.0/16 also reserved)
- Organization-local scope (239.192.0.0 to 239.251.255.255)



Layer 2 Multicast Addressing

IP Multicast MAC Address Mapping Ethernet

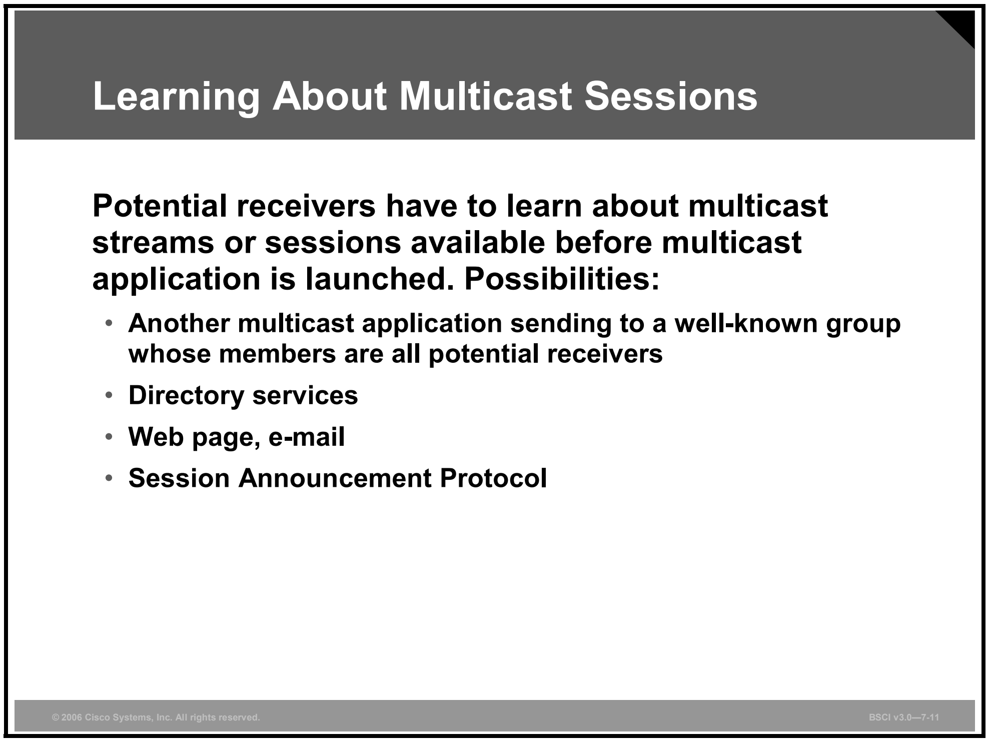
© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—7-10

The translation between IP multicast and MAC address is achieved by the mapping of the low-order 23 bits of the IP (Layer 3) multicast address into the low-order 23 bits of the IEEE (Layer 2) MAC address. In the MAC address, the low-order bit (0x01) in the first octet indicates that this packet is a Layer 2 multicast packet. The 0x01005e prefix (vendor code) has been reserved for use in mapping Layer 3 IP multicast addresses into Layer 2 MAC addresses.

Because there are 28 bits of unique address space for an IP multicast address (32 minus the first four bits containing the 1110 Class D prefix), and there are only 23 bits mapped into the IEEE MAC address, there are five bits of overlap, or 28 – 23 = 5, and 2^5 = 32. So there is a 32:1 overlap of Layer 3 addresses to Layer 2 addresses. Therefore, be aware that several Layer 3 addresses may map to the same Layer 2 multicast address.

For example, all the IP multicast addresses in this table map to the same Layer 2 multicast of 01-00-5e-0a-00-01.

224.10.0.1	225.10.0.1	226.10.0.1	227.10.0.1	228.10.0.1
229.10.0.1	230.10.0.1	231.10.0.1	232.10.0.1	233.10.0.1
234.10.0.1	235.10.0.1	236.10.0.1	237.10.0.1	238.10.0.1
239.10.0.1	224.138.0.1	225.138.0.1	226.138.0.1	227.138.0.1
228.138.0.1	229.138.0.1	230.138.0.1	231.138.0.1	232.138.0.1
233.138.0.1	234.138.0.1	235.138.0.1	236.138.0.1	237.138.0.1
238.138.0.1	239.138.0.1			



Learning About Multicast Sessions

Potential receivers have to learn about multicast streams or sessions available before multicast application is launched. Possibilities:

- **Another multicast application sending to a well-known group whose members are all potential receivers**
- **Directory services**
- **Web page, e-mail**
- **Session Announcement Protocol**

Whenever a multicast application is started on a receiver, the application has to know which multicast group to join. The application has to learn about the available sessions or streams, which typically map to one or more IP multicast groups.

These are several possibilities for applications to learn about the sessions:

- The application may join a well-known predefined group, to which announcements about available sessions are made.
- Some type of directory services is available, and the application may contact the appropriate directory server.
- The application may be launched from a web page on which the sessions are listed as URLs; even e-mail may be used.

The Session Directory (sd) application acts as a guide, displaying multicast content. A client application runs on a PC and lets the user know what content is available. This directory application uses either Session Description Protocol (SDP) or Session Announcement Protocol (SAP) to learn about the content. (Note that both the sd application and SDP are sometimes called “SDR” or “sdr” and that in Cisco documentation, SDP/SAP is referred to as “sdr.”)

The original sd application served as a means to announce available sessions and to assist in creating new sessions. The initial sd tool was revised, resulting in the Session Description Protocol tool (referred to in this course as “SDR”), which is an applications tool that allows the following:

- Session description and its announcement
- Transport of session announcement via well-known multicast groups (224.2.127.254)
- Creation of new sessions

At the receiver side, SDR is used to learn about available groups or sessions. If a user clicks an icon describing a multicast stream listed via SDR, a join to that multicast group is initiated.

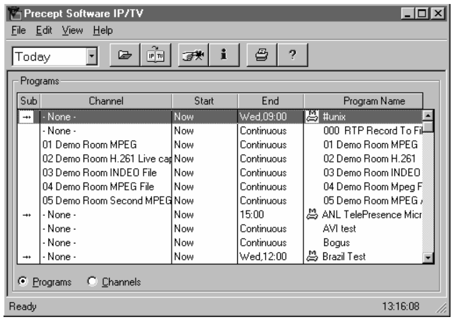
When SDR is used at the sender side, it is used to create new sessions and to avoid address conflicts. Senders at the time of session creation consult their respective SDR caches (senders are also receivers) and choose one of the unused multicast addresses. When the session is created, the senders start announcing it—with all the information that is needed by receivers to successfully join the session.

RFC 3266, which defines SDP, defines the standard set of variables that describe the sessions. Most of those variables were inherited from the SDR tool. The transport itself is not defined in this RFC. The packets describing the session may be transported across the multicast-enabled network via several mechanisms:

- SAP, defined in RFC 2974, carries the session information.
- Session Initiation Protocol (SIP), which is defined in RFC 2543, is a signaling protocol for Internet conferencing, telephony, presence, events notification, and instant messaging.
- Real Time Streaming Protocol (RTSP), which is defined in RFC 2326, serves mainly as a control protocol in a multimedia environment. RTSP allows videocassette recorder (VCR)-like controls (select, forward, rewind, pause, stop, and so on) and also carries information on a session.
- E-mail (in Multipurpose Internet Mail Extensions [MIME] format) may also carry SDR packets describing the session.
- Web pages may provide session descriptions in standardized SDR format also.

A Cisco IP/TV Example

- Cisco IP/TV application
- Clients (viewers) use program listing
  - Contact the server directly
  - Listen to SAP announcements



Sub	Channel	Start	End	Program Name
→	- None -	Now	Wed,09:00	#unix
	- None -	Now	Continuous	000 RTP Record To Fil
	01 Demo Room MPEG	Now	Continuous	01 Demo Room MPEG
	02 Demo Room H.261 Live cap	Now	Continuous	02 Demo Room H.261
	03 Demo Room INDEO File	Now	Continuous	03 Demo Room INDEO
	04 Demo Room MPEG File	Now	Continuous	04 Demo Room Mpeg F
	05 Demo Room Second MPEG	Now	Continuous	05 Demo Room MPEG /
→	- None -	Now	15:00	ANL TelePresence Micr
	- None -	Now	Continuous	AVI test
	- None -	Now	Continuous	Bogus
→	- None -	Now	Wed,12:00	Brazil Test

● Programs ○ Channels

Ready 13:16:08

The SDR mechanisms are shown in this Cisco IP/TV example. Cisco IP/TV generally has three components: server (the source), content manager (the directory server), and viewer (the receiver).

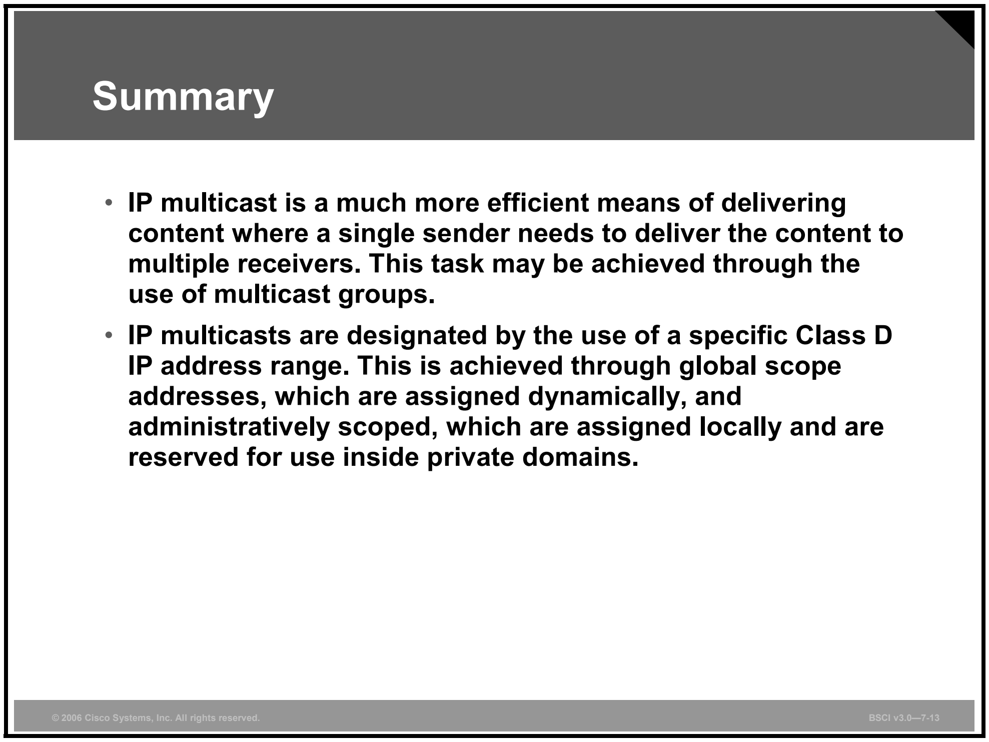
The viewers may perform either of these two actions:

- Contact the content manager directly (by unicast) and request the list of available programs (sessions, streams) from it
- Listen to periodic SAP announcements

Cisco IP/TV uses SAP to transport the SDR sessions to the viewer. The standard SDR format for session description is used.

Summary

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



Summary

- IP multicast is a much more efficient means of delivering content where a single sender needs to deliver the content to multiple receivers. This task may be achieved through the use of multicast groups.
- IP multicasts are designated by the use of a specific Class D IP address range. This is achieved through global scope addresses, which are assigned dynamically, and administratively scoped, which are assigned locally and are reserved for use inside private domains.

Lesson 2

IGMP and Layer 2 Issues

Overview

This module introduces Internet Group Management Protocol (IGMP), which has evolved through three versions (1, 2, and 3). Understanding this protocol is fundamental in defining the multicast group membership join and leave process, which is a required function of multicasting.

Without control, multicast packets are flooded as unknown unicast frames by an Ethernet switch. IGMP snooping and Cisco Group Management Protocol (CGMP) are used to solve this problem.

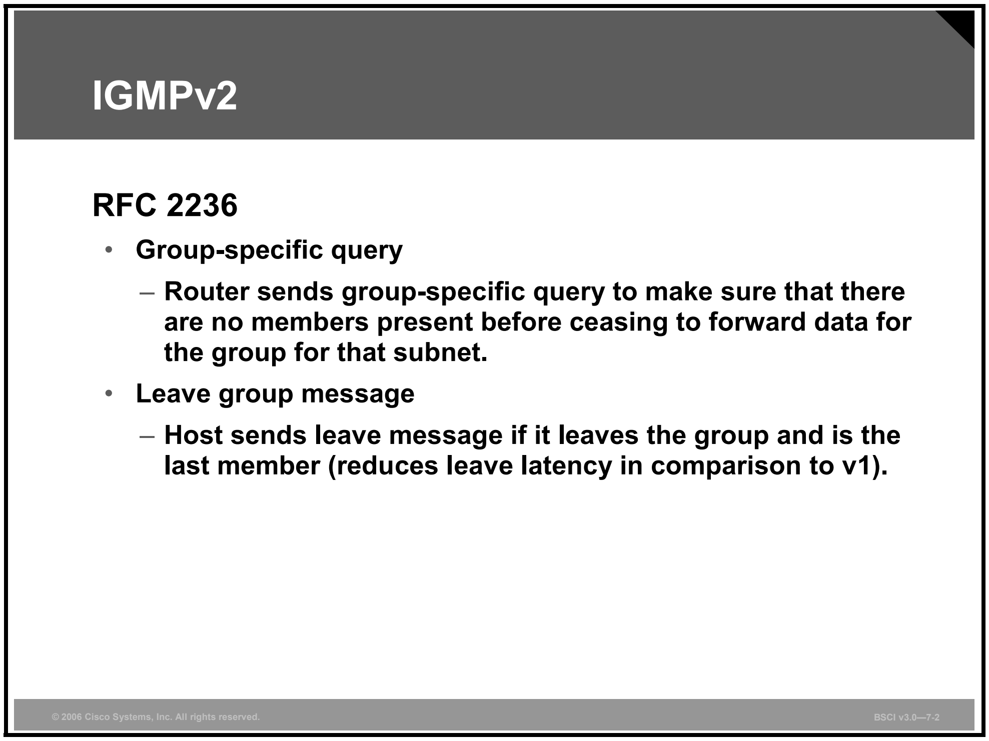
Objectives

Upon completing this lesson, you will be able to define and implement IGMP and resolve frame-forwarding issues in Ethernet switching. This ability includes being able to meet these objectives:

- Explain how IGMPv2 manages which groups are active or inactive on a particular subnet
- Explain how IGMPv3 manages which groups are active or inactive on a particular subnet
- Explain the methods used to deal with multicast in a Layer 2 switching environment
- Describe CGMP operation
- Describe IGMP snooping operation

Introducing IGMPv2

IGMP version 2 (IGMPv2) offers several enhancements to IGMP version 1 (IGMPv1), including a leave message and explicit poll.



IGMPv2

RFC 2236

- **Group-specific query**
  - **Router sends group-specific query to make sure that there are no members present before ceasing to forward data for the group for that subnet.**
- **Leave group message**
  - **Host sends leave message if it leaves the group and is the last member (reduces leave latency in comparison to v1).**

IGMP is a host-to-router protocol used when hosts want to join a multicast group. With IGMPv1, routers send periodic membership queries to the multicast address 224.0.0.1. Hosts send membership reports to the group multicast address they want to join; hosts silently leave the multicast group.

In response to some of the limitations discovered in IGMPv1, work was begun on IGMPv2. Most of the changes between IGMPv1 and IGMPv2 were made primarily to address the issues of leave and join latencies, as well as to address ambiguities in the original protocol specification.

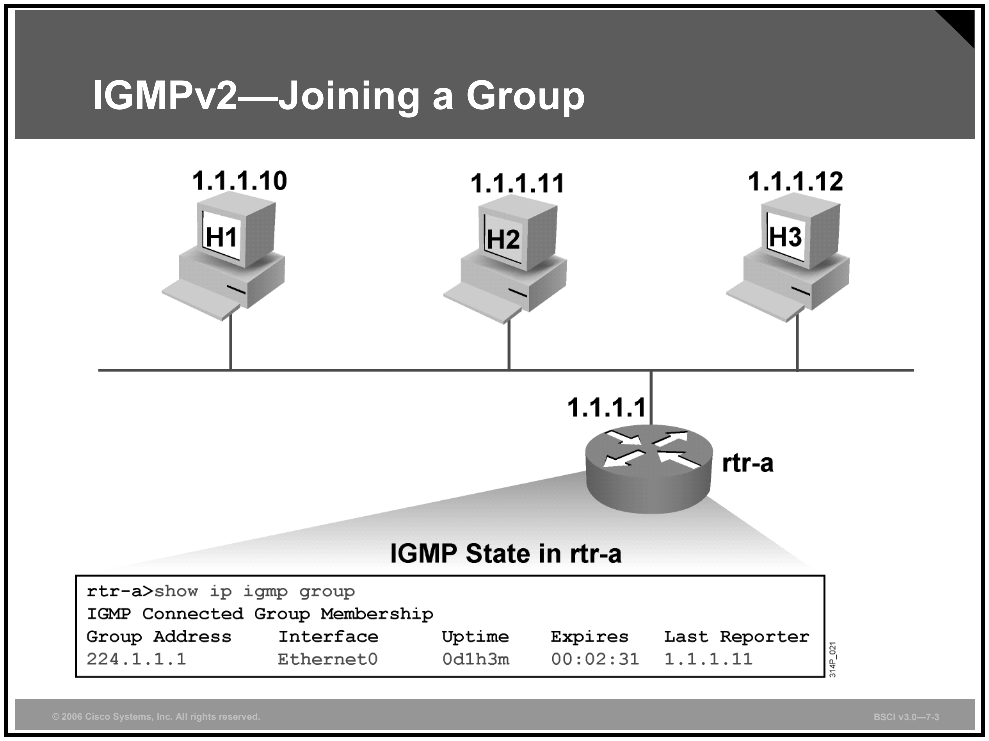
These were some important changes made in revising IGMPv1 to IGMPv2:

- Group-specific queries
- Leave group message
- Querier election mechanism
- Query-interval response time

A group-specific query that was added in IGMPv2 allows the router to query membership in a single group instead of in all groups, which is an optimized way to find out whether any members are left in a group without asking all groups for a report. The difference between the group-specific query and the membership query is that a membership query is multicast to the all-hosts (224.0.0.1) address, while a group-specific query for group G, for example, is multicast to the group G multicast address.

A leave group message allows hosts to tell the router that they are leaving the group. This information reduces the leave latency for the group on the segment when the member who is leaving is the last member of the group. The standard is written defining the time when leave group messages must be sent.

The query-interval response time was added to control the burstiness of reports. This time is set in queries to convey to the members how much time they have to respond to a query with a report. IGMPv2 is backward-compatible with IGMPv1.



IGMPv2—Joining a Group

IGMP State in rtr-a

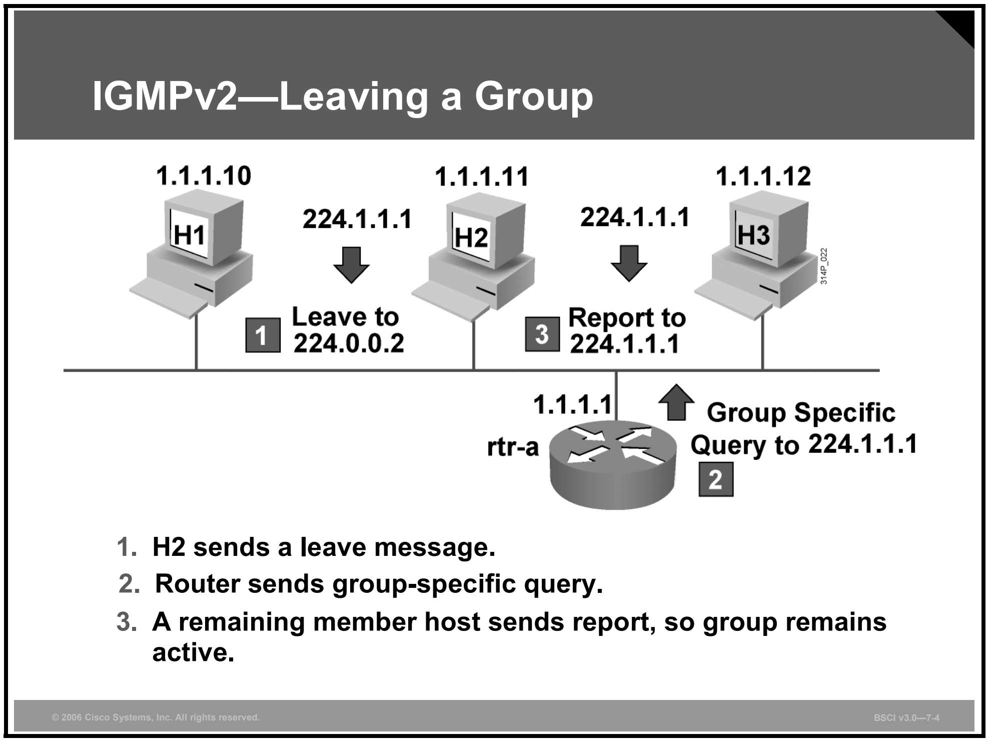
```
rtr-a>show ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0      0d1h3m    00:02:31  1.1.1.11
```

Members joining a multicast group do not have to wait for a query to join. They send an unsolicited report indicating their interest. This procedure reduces join latency for the end system joining if no other members are present.

In this example, after host H2 sends the join group message for group 224.1.1.1, group 224.1.1.1 is active on the router interface Ethernet0. Using the **show ip igmp group** command reveals the following:

- Group 224.1.1.1 has been active on this interface for 1 hour and 3 minutes.
- Group 224.1.1.1 expires (and is deleted) in 2 minutes and 31 seconds if an IGMP host membership report for this group is not heard in that time.
- The last host to report membership was 1.1.1.11 (H2).

When there are two IGMP routers on the same Ethernet segment (broadcast domain), the router with the highest IP address is the designated querier.



IGMPv2—Leaving a Group

1. H2 sends a leave message.
2. Router sends group-specific query.
3. A remaining member host sends report, so group remains active.

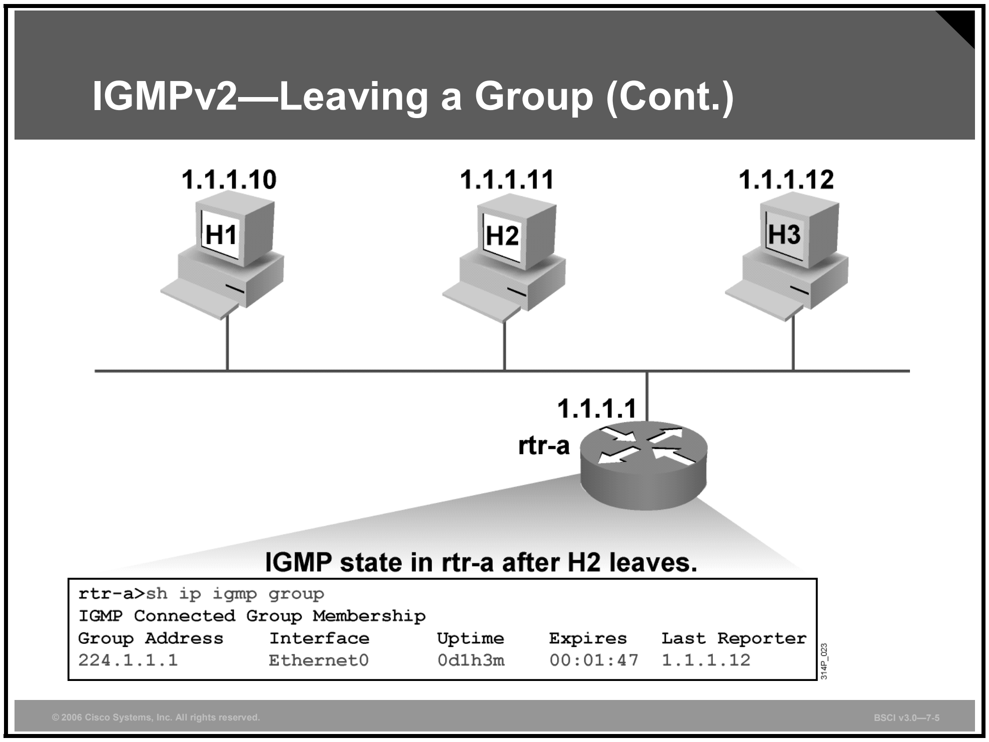
BSCI v3.0—7-4

In IGMPv1, hosts leave passively. They do not explicitly say that they are leaving; they just stop reporting their membership by responding to membership queries. However, IGMPv2 has explicit leave group messages.

When the IGMPv2 router receives a leave message, it responds by sending a group-specific query for the associated group to see whether there are other hosts interested in receiving traffic for the group. This process helps to reduce overall leave latency.

In the example shown, hosts H2 and H3 are members of multicast group 224.1.1.1. At a certain point, host H2 leaves the group and announces the departure by sending a leave message to multicast group 224.0.0.2 (all multicast routers).

The router hears the leave message and sends a group-specific query to see if any other group members are present. Host H3 has not left the multicast group 224.1.1.1 yet, so it responds with a report message. This response tells the router to keep sending multicast for 224.1.1.1, because there is still at least one member present.



IGMPv2—Leaving a Group (Cont.)

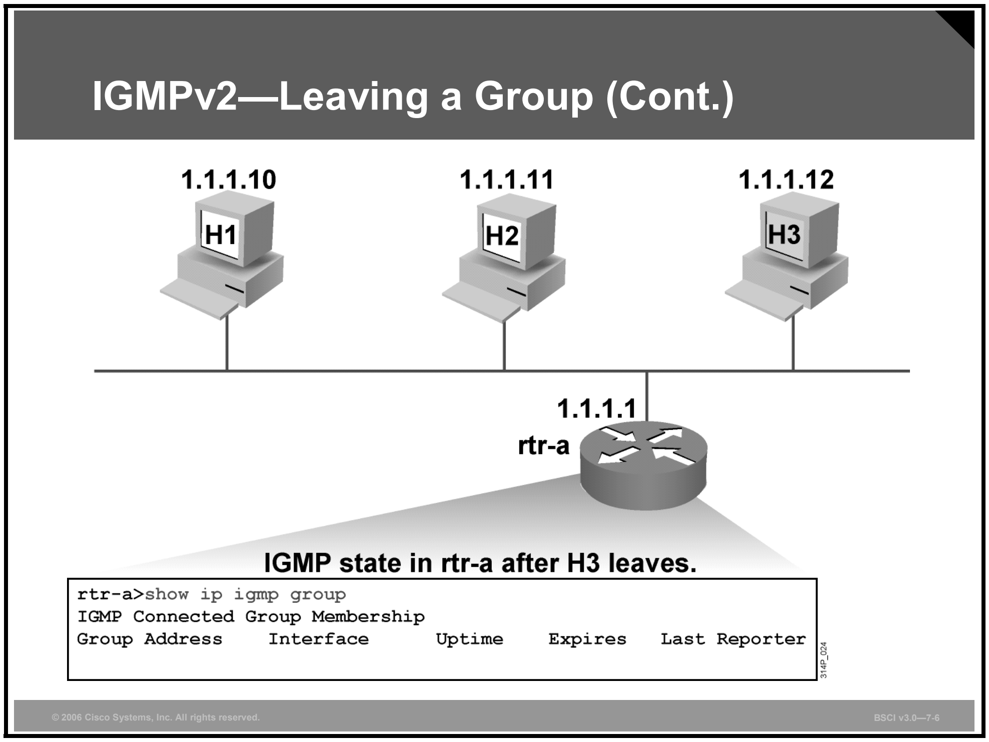
IGMP state in rtr-a after H2 leaves.

```
rtr-a>sh ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0      0d1h3m    00:01:47  1.1.1.12
```

In the example in the figure, the multicast group 224.1.1.1 is still active. However, the IGMP information shows that host H3 is the last host to send an IGMP group membership report.

After receiving a leave message from H3, the router sends a group-specific query to see whether any other group members are present.

Because host H3 was the last remaining member of the multicast group 224.1.1.1, no IGMP membership report for group 224.1.1.1 is received, and the group times out. This activity typically takes from 1 to 3 seconds, from the time that the leave message is sent until the group-specific query times out and multicast traffic stops flowing for that group.



IGMPv2—Leaving a Group (Cont.)

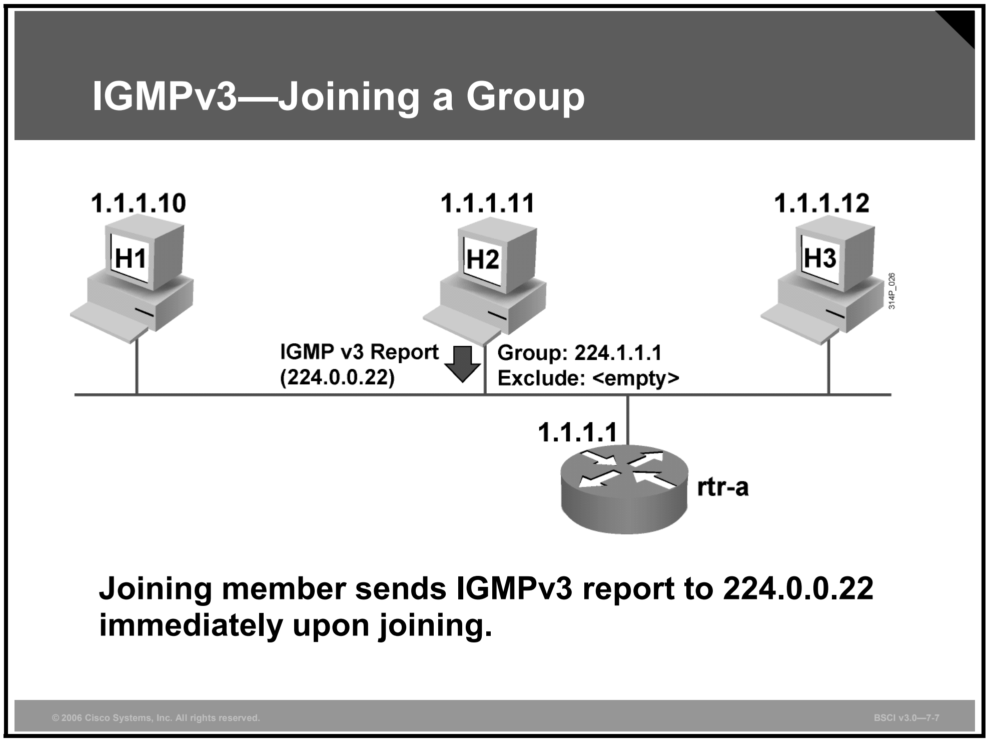
IGMP state in rtr-a after H3 leaves.

```
rtr-a>show ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires   Last Reporter
```

In the example in the figure, all hosts have left the 224.1.1.1 group on Ethernet0. This status is indicated in the output of the **show ip igmp group** command.

Introducing IGMPv3

IGMP version 3 (IGMPv3) adds the ability to filter multicasts based on multicast source.



IGMPv3—Joining a Group

1.1.1.10

H1

1.1.1.11

H2

1.1.1.12

H3

IGMP v3 Report
(224.0.0.22)

Group: 224.1.1.1
Exclude: <empty>

1.1.1.1

rtr-a

314P_026

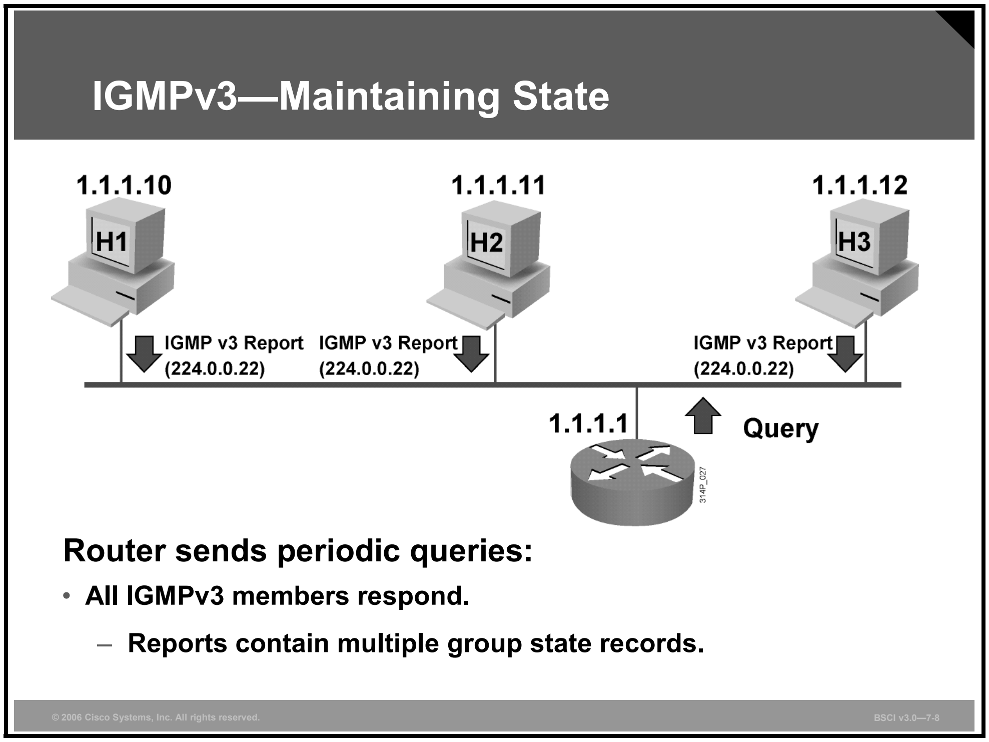
Joining member sends IGMPv3 report to 224.0.0.22 immediately upon joining.

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—7-7

The main intention of IGMPv3, which is a proposed standard, is to allow hosts to indicate that they want to receive traffic only from particular sources within a multicast group.

This enhancement makes the utilization of routing resources more efficient.



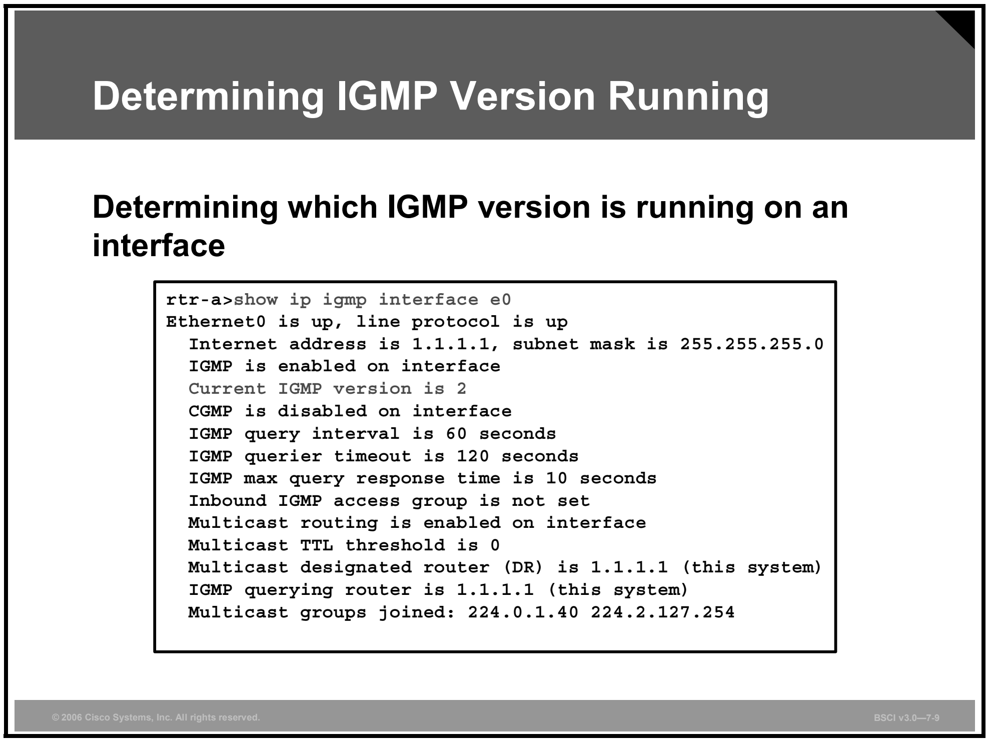
IGMPv3—Maintaining State

Router sends periodic queries:

- **All IGMPv3 members respond.**
  - **Reports contain multiple group state records.**

In this example, which shows IGMPv3 operation, host H3 sends a join message with an explicit request for join to sources in the source list. This is the list that IGMPv3 uses for "source filtering"—that is, the ability for a system to report interest in receiving packets *only* from specific source addresses, or from *all but* specific source addresses, sent to a particular multicast address. That information may be used by multicast routing protocols to avoid delivering multicast packets from specific sources to networks where there are no interested receivers.

In IGMPv3, reports are sent to 224.0.0.22 rather than 224.0.0.2.



Determining IGMP Version Running

Determining which IGMP version is running on an interface

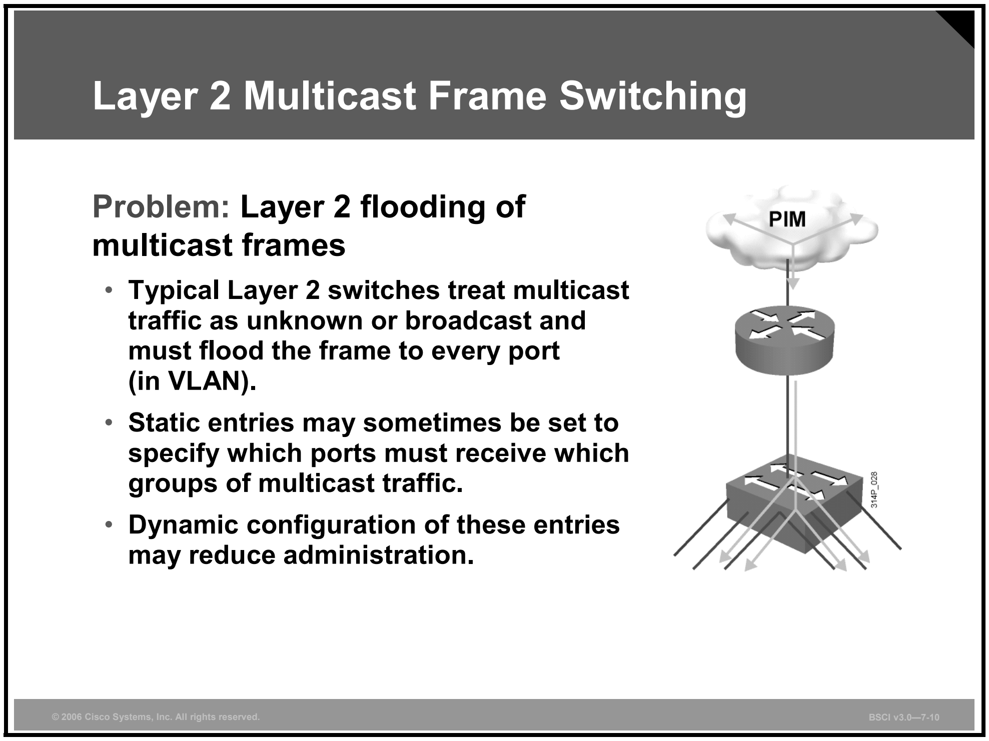
```
rtr-a>show ip igmp interface e0
Ethernet0 is up, line protocol is up
  Internet address is 1.1.1.1, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Inbound IGMP access group is not set
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 1.1.1.1 (this system)
  IGMP querying router is 1.1.1.1 (this system)
  Multicast groups joined: 224.0.1.40 224.2.127.254
```

Use the **show ip igmp interface** command to determine which version of IGMP is currently active on an interface.

The IGMP version is indicated by the line in the example that reads “Current IGMP version is 2.”

Multicast in Layer 2 Switching

This topic explains the methods used to deal with multicast in a Layer 2 switching environment.



Layer 2 Multicast Frame Switching

Problem: Layer 2 flooding of multicast frames

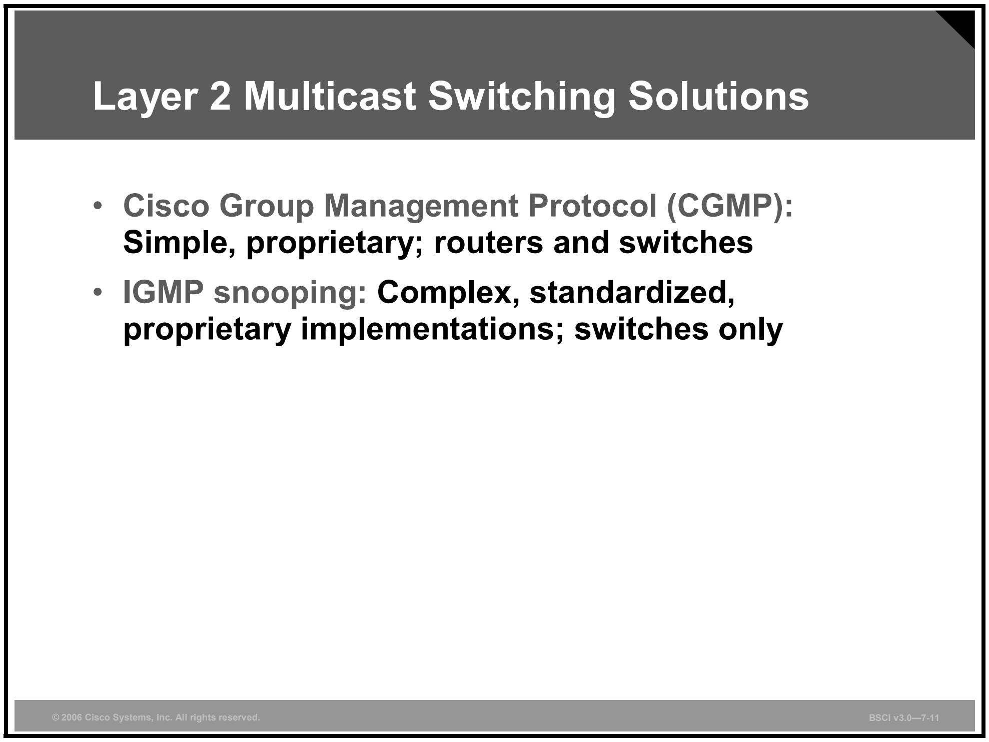
- **Typical Layer 2 switches treat multicast traffic as unknown or broadcast and must flood the frame to every port (in VLAN).**
- **Static entries may sometimes be set to specify which ports must receive which groups of multicast traffic.**
- **Dynamic configuration of these entries may reduce administration.**

BSCI v3.0—7-10

For most Layer 2 switches, multicast traffic is normally treated like an unknown MAC address or broadcast frame that causes the frame to be flooded out every port within a VLAN. This treatment is acceptable for unknowns and broadcasts, but, as noted earlier, IP multicast hosts may join and be interested in only specific multicast groups. On most Layer 2 switches, all of this traffic is forwarded out of all ports, resulting in wasted bandwidth on both of the segments and on the end stations.

One method that Cisco Catalyst switches used to circumvent this is to allow the administrator to configure the switch to manually associate a multicast MAC address with various ports. For example, the administrator may configure ports 5, 6, and 7 so that only ports 5, 6, and 7 will receive the multicast traffic destined for the multicast group. This method works but is not scalable. IP multicast hosts dynamically join and leave groups using IGMP to signal to the multicast router. Dynamic configuration of the forwarding tables of the switches may be more effective and may reduce user administration.

At layer 3, the router may also multicast over the Internet using **Protocol-Independent Multicast (PIM).** This is a family of multicast routing protocols that can provide one-to-many and many-to-many distribution of data over the Internet. The "protocol-independent" part refers to the fact that PIM does not include its own topology discovery mechanism, but instead uses routing information supplied by other traditional routing protocols such as BGP.



Layer 2 Multicast Switching Solutions

- **Cisco Group Management Protocol (CGMP): Simple, proprietary; routers and switches**
- **IGMP snooping: Complex, standardized, proprietary implementations; switches only**

BSCI v3.0—7-11

To improve the behavior of the switches when they receive multicast frames, many multicast switching solutions have been developed, such as the following:

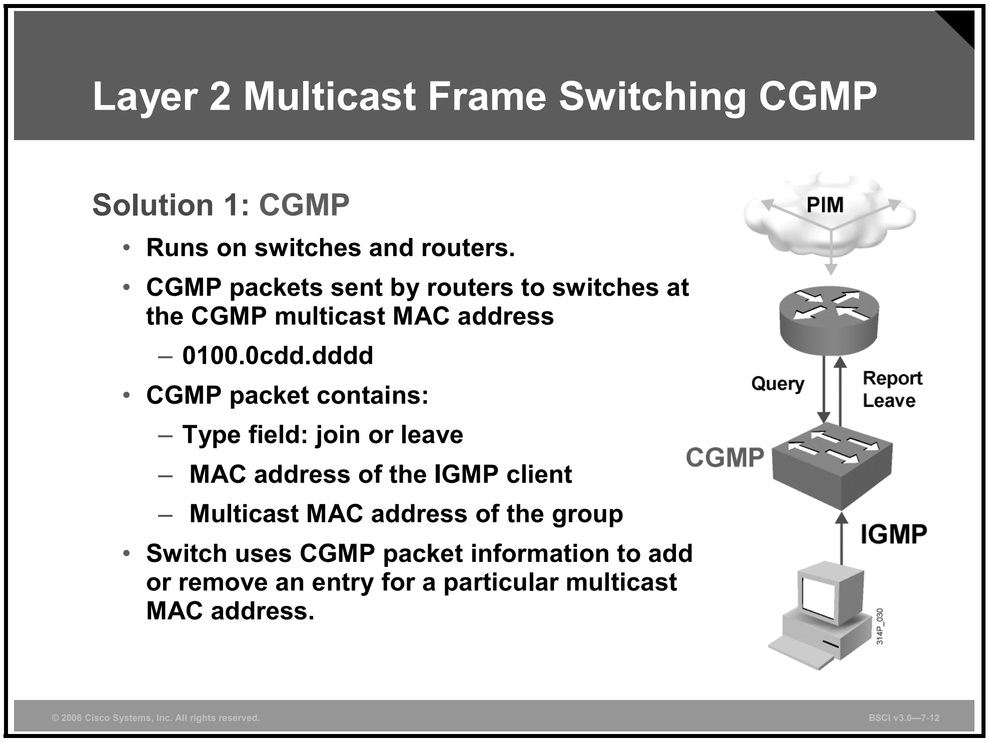
- CGMP
- IGMP snooping

CGMP is a Cisco Systems proprietary protocol that runs between a multicast router and a switch. This protocol enables the Cisco multicast router, following the receipt of IGMP messages sent by hosts, to inform the switch about the information contained in the IGMP packet.

With IGMP snooping, the switch intercepts IGMP messages from the host and updates the MAC table accordingly. To implement IGMP snooping without suffering switch performance loss, it is necessary to make the switch Layer 3-aware. This result is typically accomplished by using Layer 3 ASICs.

Cisco Group Management Protocol

This topic describes CGMP and how to enable it.



Layer 2 Multicast Frame Switching CGMP

Solution 1: CGMP

- **Runs on switches and routers.**
- **CGMP packets sent by routers to switches at the CGMP multicast MAC address**
  - **0100.0cdd.dddd**
- **CGMP packet contains:**
  - **Type field: join or leave**
  - **MAC address of the IGMP client**
  - **Multicast MAC address of the group**
- **Switch uses CGMP packet information to add or remove an entry for a particular multicast MAC address.**

BSCI v3.0—7-12

CGMP is the most common multicast switching solution designed by Cisco.

CGMP is based on a client/server model, where the router may be considered a CGMP server and the switch assumes the client role. There are software components running on both devices, with the router translating IGMP messages into CGMP commands, which are then processed in the switches and used to populate the Layer 2 forwarding tables with the correct multicast entries.

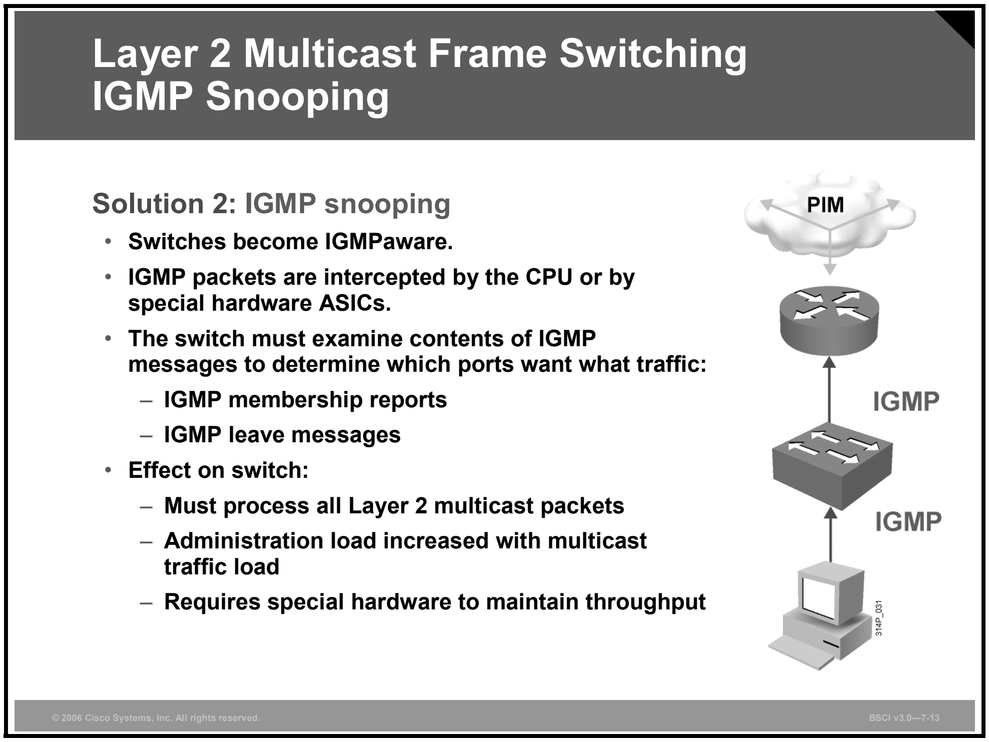
The basis of CGMP is that the IP multicast router sees all IGMP packets and, therefore, may inform the switch when specific hosts join or leave multicast groups. Routers use well-known CGMP multicast MAC addresses to send CGMP control packets to the switch. The switch then uses this information to program the forwarding table.

When the router sees an IGMP control packet, it creates a CGMP packet that contains the request type (join or leave), the Layer 2 multicast MAC address, and the actual MAC address of the client.

This packet is sent to the well-known CGMP multicast MAC address 0x0100.0cdd.dddd to which all CGMP switches listen. The CGMP control message is then interpreted, and the proper entries are created in the switch content-addressable memory (CAM) table to constrain the forwarding of multicast traffic for this group.

IGMP Snooping

This topic describes IGMP snooping operation.



Layer 2 Multicast Frame Switching IGMP Snooping

Solution 2: IGMP snooping

- **Switches become IGMPaware.**
- **IGMP packets are intercepted by the CPU or by special hardware ASICs.**
- **The switch must examine contents of IGMP messages to determine which ports want what traffic:**
  - **IGMP membership reports**
  - **IGMP leave messages**
- **Effect on switch:**
  - **Must process all Layer 2 multicast packets**
  - **Administration load increased with multicast traffic load**
  - **Requires special hardware to maintain throughput**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—7-13

The second multicast switching solution is IGMP snooping.

As its name implies, switches become IGMP-aware and listen in on the IGMP conversations between hosts and routers.

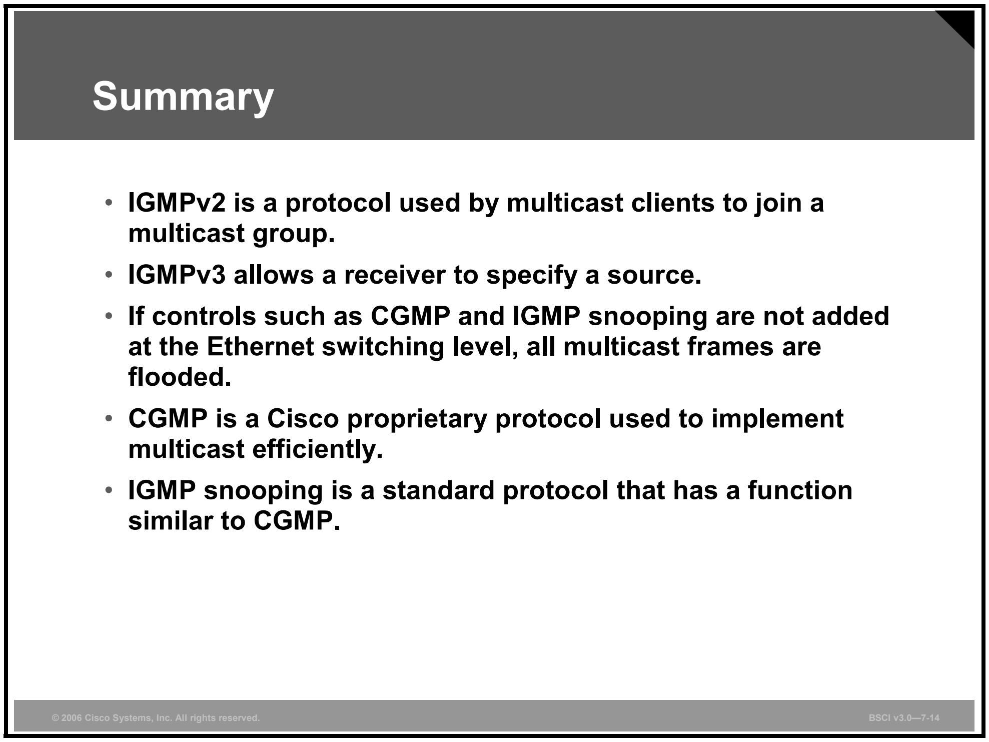
This activity requires the processor in each switch to identify and intercept a copy of all IGMP packets flowing between routers and hosts and vice versa. It includes these IGMP packets:

- IGMP membership reports
- IGMP leaves

If care is not taken as to how IGMP snooping is implemented, a switch may have to intercept all Layer 2 multicast packets to identify IGMP packets. This action may have a significant impact on switch performance. Proper designs require special hardware (Layer 3 ASICs) to avoid this problem, which may directly affect the overall cost of the switch. Thus, switches must effectively become Layer 3-aware to avoid serious performance problems because of IGMP snooping.

Summary

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



Summary

- IGMPv2 is a protocol used by multicast clients to join a multicast group.
- IGMPv3 allows a receiver to specify a source.
- If controls such as CGMP and IGMP snooping are not added at the Ethernet switching level, all multicast frames are flooded.
- CGMP is a Cisco proprietary protocol used to implement multicast efficiently.
- IGMP snooping is a standard protocol that has a function similar to CGMP.

BSCI v3.0—7-14

Lesson 3

Explaining Multicast Routing Protocols

Overview

The focus of this lesson is on the accurate and efficient construction of multicast distribution trees. The tree is then utilized for proper multicast packet forwarding using a multicast routing protocol, Protocol Independent Multicast (PIM).

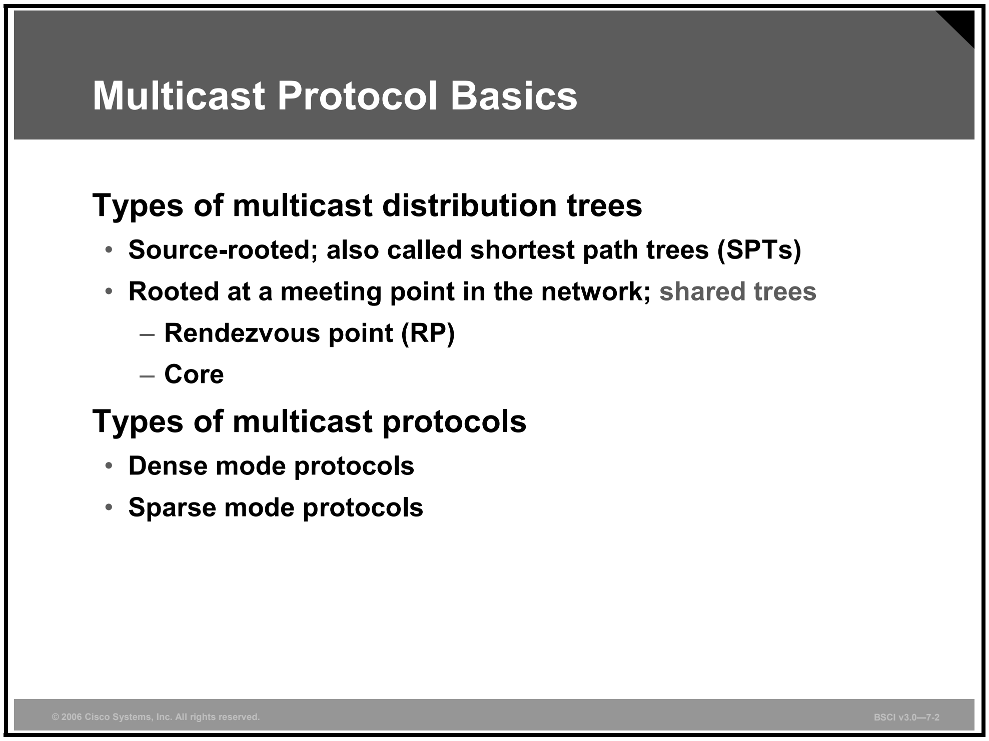
Objectives

Upon completing this lesson, you will be able to describe and select multicast routing protocols. This ability includes being able to meet these objectives:

- Describe a multicast network in terms of the IP routing protocols used over various segments
- Describe multicast distribution trees: source trees and shared trees
- Explain IP multicast routing
- Explain how each of the PIM modes forwards multicast traffic
- Describe PIM-DM
- Describe PIM-SM

Protocols Used in Multicast

This topic describes a multicast network in terms of the IP multicast routing protocols and processes used over various segments.



Multicast Protocol Basics

Types of multicast distribution trees

- **Source-rooted; also called shortest path trees (SPTs)**
- **Rooted at a meeting point in the network; shared trees**
  - **Rendezvous point (RP)**
  - **Core**

Types of multicast protocols

- **Dense mode protocols**
- **Sparse mode protocols**

BSCI v3.0—7-2

Multicast distribution trees define the path from the source to the receivers, over which the multicast traffic flows.

There are two types of multicast distribution trees—source-rooted or shortest path trees (SPTs) and shared trees.

With a source-rooted tree, a separate tree is built for each source to all members of its group. Because the source-rooted tree takes a direct, or the shortest, path from source to its receivers, it is also called an SPT.

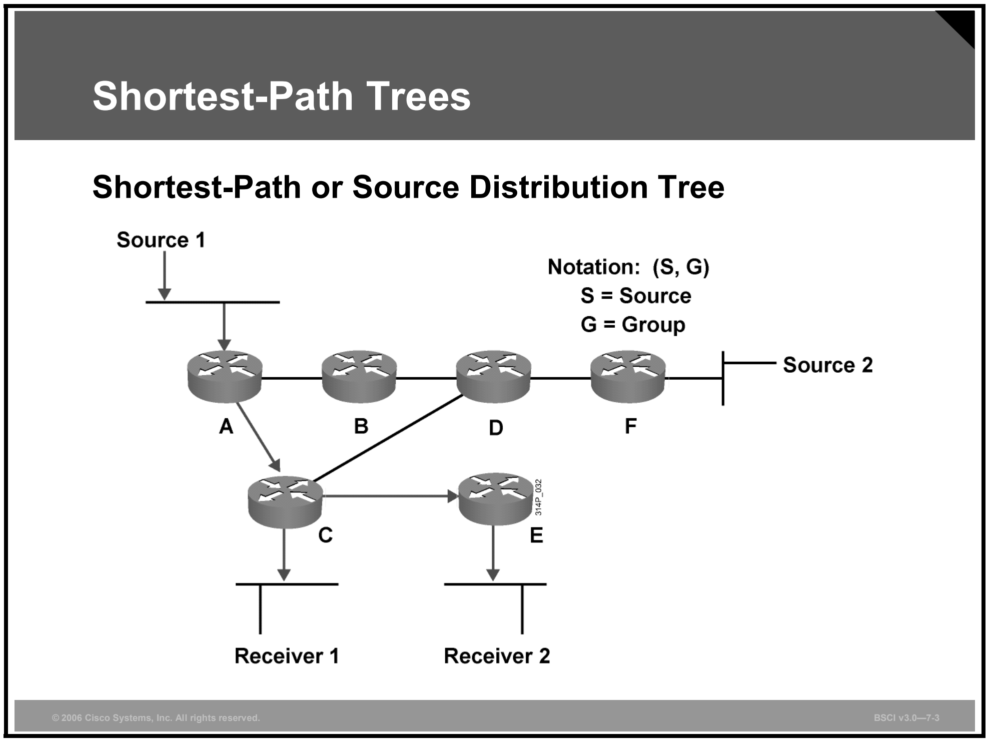
Shared tree protocols create multicast forwarding paths that rely on a central core router that serves as a rendezvous point (RP) between multicast sources and destinations. Sources initially send their multicast packets to the RP, which in turn forwards data through a shared tree to the members of the group. A shared tree is less efficient than an SPT (paths between the source and receivers are not necessarily the shortest), but it is less demanding on routers (memory, CPU).

There are basically two types of multicast routing protocols: dense mode protocols and sparse mode protocols:

- Dense mode protocols flood multicast traffic to all parts of the network and prune the flows where there are no receivers using a periodic flood-and-prune mechanism.
- Sparse mode protocols use an explicit join mechanism where distribution trees are built on demand by explicit tree join messages sent by routers that have directly connected receivers.

Multicast Distribution Trees

This topic describes multicast distribution trees: SPTs and shared trees.



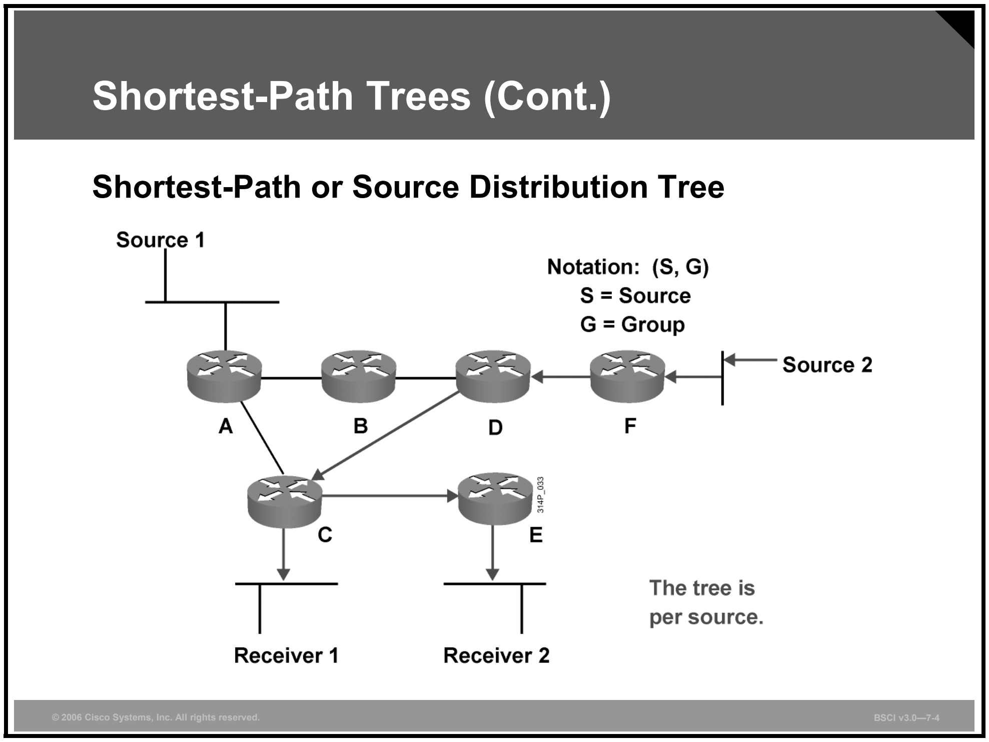
Shortest-Path Trees

Shortest-Path or Source Distribution Tree

Notation: (S, G)
S = Source
G = Group

The example in the figure shows an SPT between source 1 and receiver 1 and receiver 2. It is appropriately assumed that the path between the source and receivers over routers A, C, and E is the path with the lowest cost.

Packets are forwarded according to source and group address pair down the SPT. For this reason, the forwarding state associated with the SPT is referred to by the notation "(S, G)" (pronounced "S comma G"), where S is the IP address of the source, and G is the multicast group address.



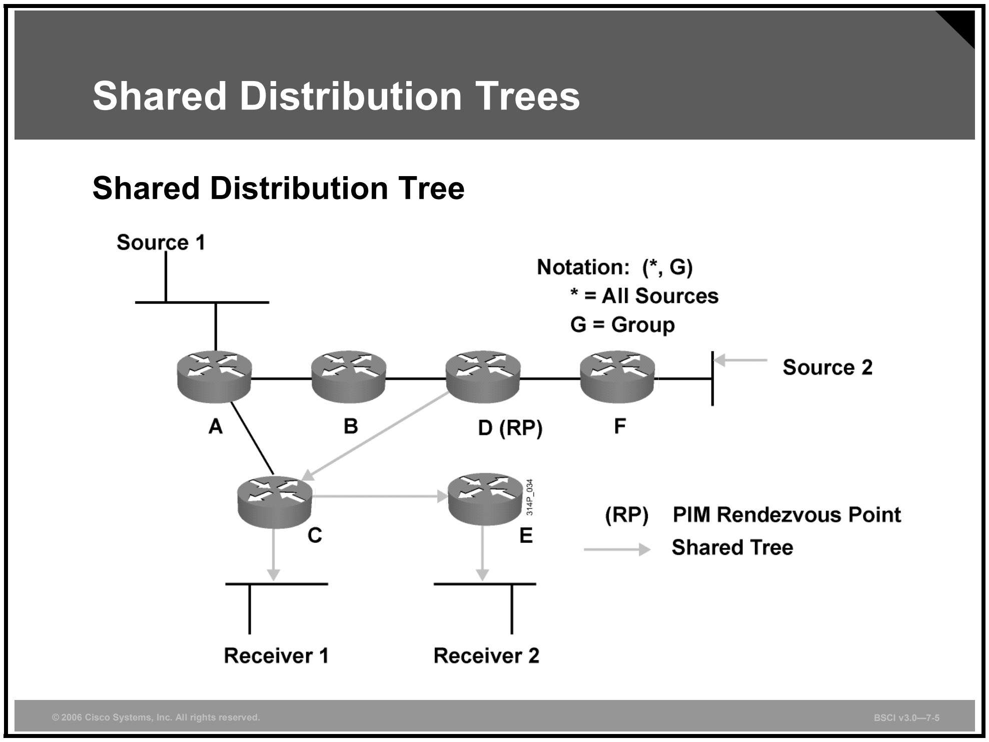
Shortest-Path Trees (Cont.)

Shortest-Path or Source Distribution Tree

Notation: (S, G)
S = Source
G = Group

The tree is per source.

The example in the figure shows another example of SPT, where source 2 is active and is sending multicast packets to receiver 1 and receiver 2. Clearly, a separate SPT is built for this purpose, this time with source 2 at the root of the SPT. The main point is that a separate SPT is built for every source S sending to group G.



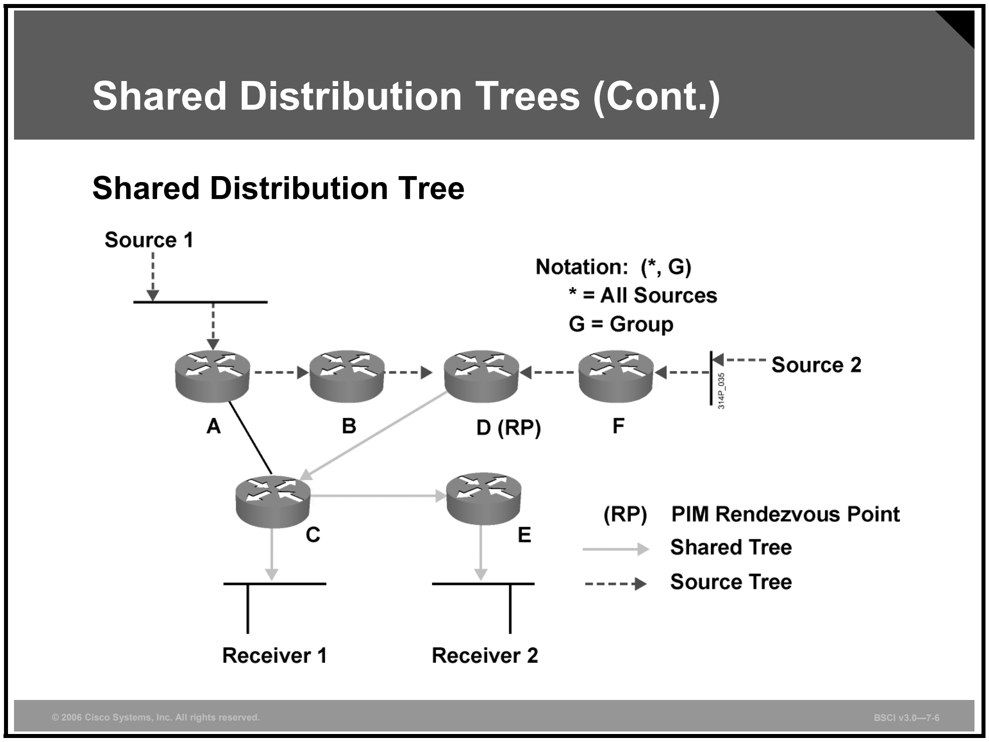
Shared Distribution Trees

Shared Distribution Tree

BSCI v3.0—7-5

The example in the figure shows a shared distribution tree. Router D is the root of this shared tree, which is built from router D to routers C and E toward receiver 1 and receiver 2. In PIM, the root of the shared tree is called an RP.

Packets are forwarded down the shared distribution tree to the receivers. The default forwarding state for the shared tree is identified by the notation "(*, G)" (pronounced "star comma G"), where the asterisk (*) is a wildcard entry, meaning any source, and G is the multicast group address.

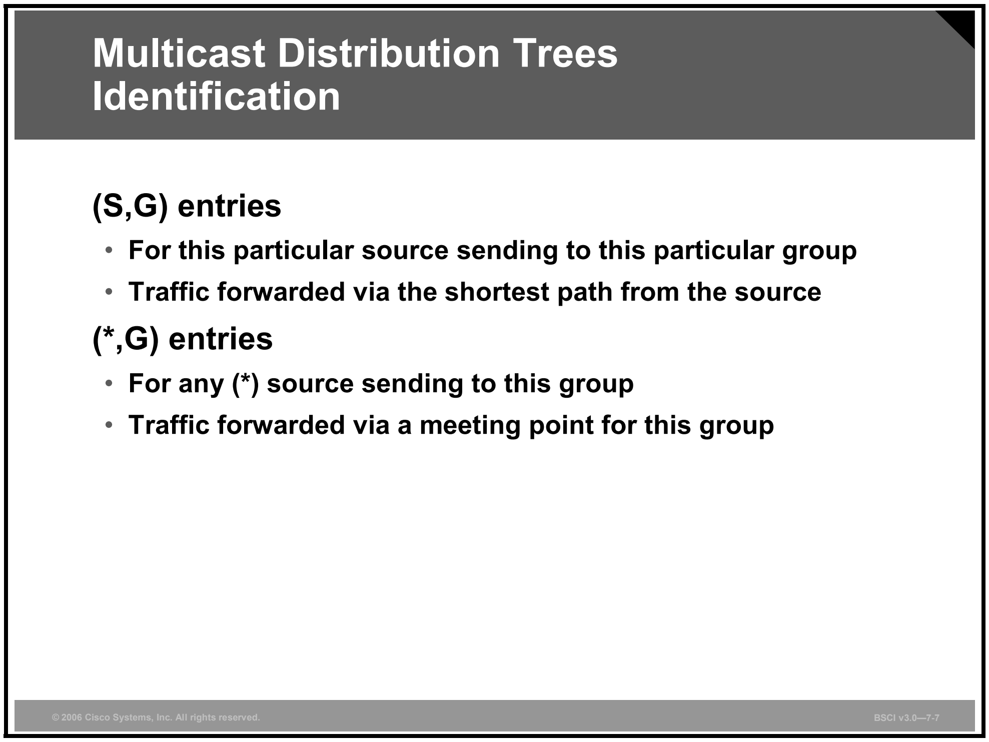


Shared Distribution Trees (Cont.)

Shared Distribution Tree

BSCI v3.0—7-6

In the example in the figure of a shared distribution tree, source 1 and source 2 are sending multicast packets toward a rendezvous point via SPTs, and from the RP, the multicast packets are flowing via a shared distribution tree toward receiver 1 and receiver 2.



Multicast Distribution Trees Identification

(S,G) entries

- **For this particular source sending to this particular group**
- **Traffic forwarded via the shortest path from the source**

(*,G) entries

- **For any (*) source sending to this group**
- **Traffic forwarded via a meeting point for this group**

BSCI v3.0—7-7

The multicast forwarding entries that appear in multicast forwarding tables may be read in the following way:

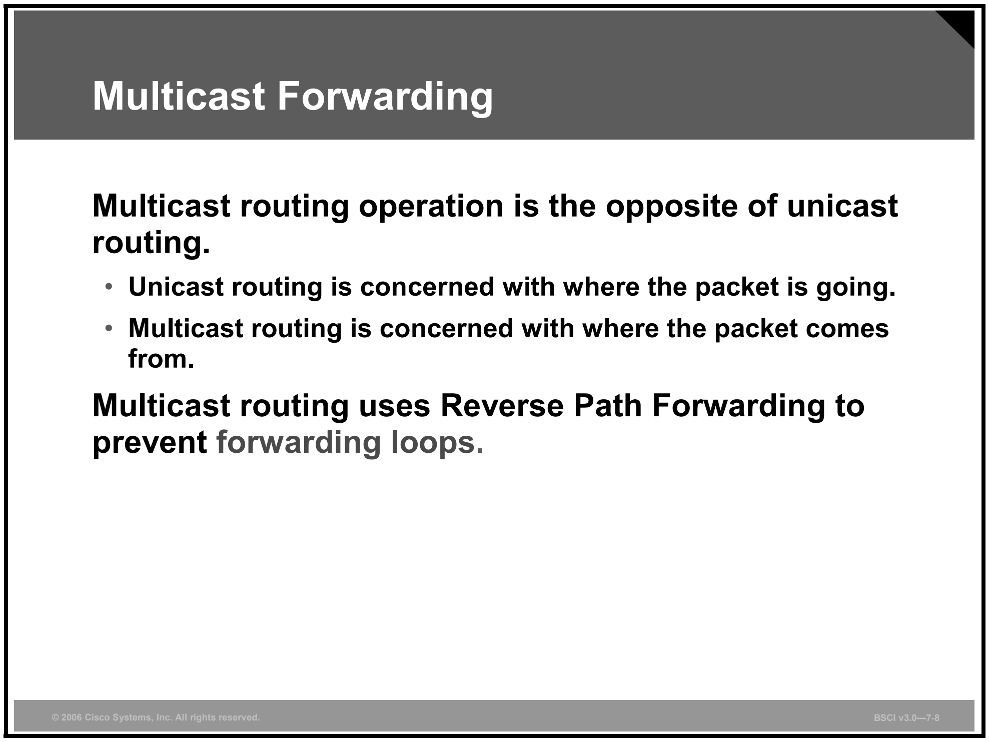
- **(S, G):** For the source S sending to the group G; those entries typically reflect the SPT but may also appear on a shared tree.
- **(*, G):** For any source (*) sending to the group G; those entries reflect the shared tree but are also created (in Cisco routers) for any existing (S, G) entry.

SPT state entries use more router memory because there is an entry for each sender and group pair, but the traffic is sent over the optimal path to each receiver, thus minimizing the delay in packet delivery.

Shared distribution tree state entries consume less router memory, but you may get suboptimal paths from a source to receivers, thus introducing extra delay in packet delivery.

Introducing IP Multicast Routing

This topic explains IP multicast routing.



Multicast Forwarding

Multicast routing operation is the opposite of unicast routing.

- **Unicast routing is concerned with where the packet is going.**
- **Multicast routing is concerned with where the packet comes from.**

Multicast routing uses Reverse Path Forwarding to prevent forwarding loops.

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—7-8

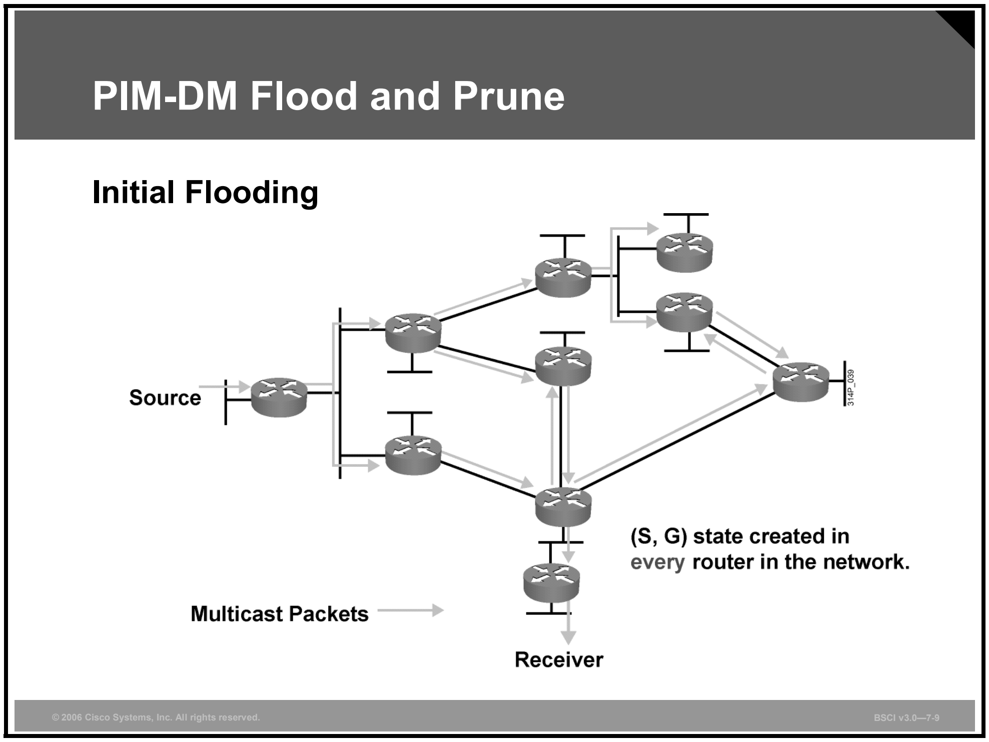
In unicast routing, when the router receives the packet, the decision about where to forward the packet depends on the destination address of the packet. In multicast routing, the decision about where to forward the multicast packet depends on where the packet came from.

Multicast routers must know the origin of the packet, rather than its destination, which is the opposite of unicast routing. In multicast origination, the IP address denotes the known source, and the destination IP address denotes a group of unknown receivers.

Multicast routing uses a mechanism called Reverse Path Forwarding (RPF) to prevent forwarding loops and to ensure the shortest path from the source to the receivers.

Introducing PIM

This topic explains how each of the PIM modes forwards multicast traffic.



PIM-DM Flood and Prune

Initial Flooding

(S, G) state created in every router in the network.

BSCI v3.0—7-9

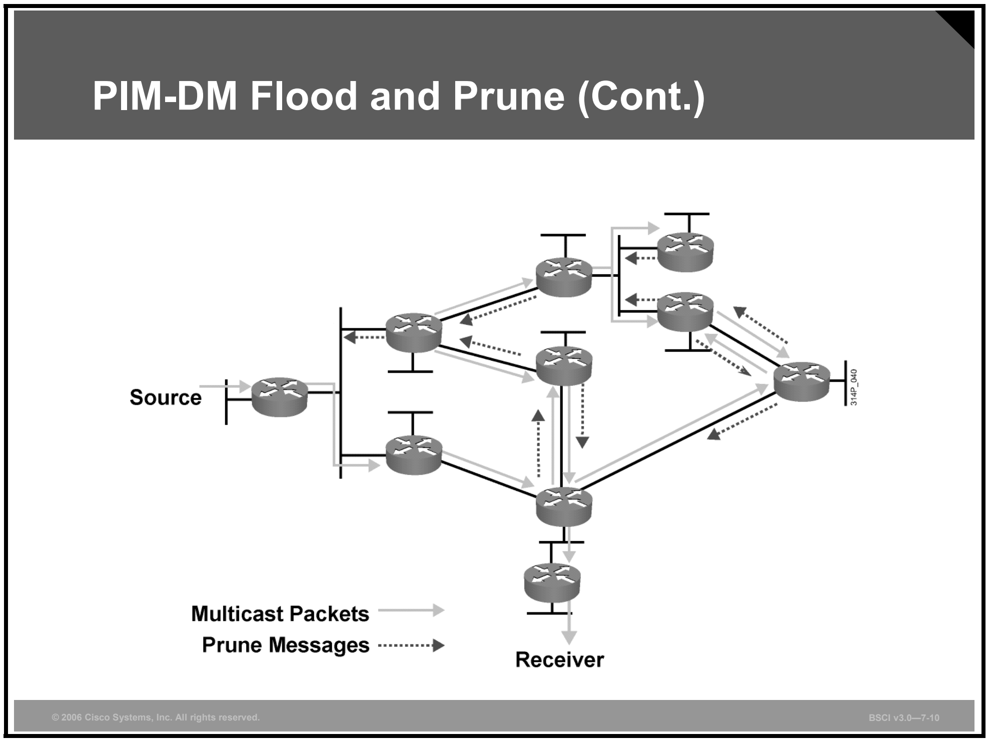
PIM dense mode (PIM-DM) initially floods multicast traffic to all parts of the network. PIM-DM initially floods traffic out of all non-RPF interfaces where there is another PIM-DM neighbor or a directly connected member of the group.

In the example in the figure, multicast traffic being sent by the source is flooded throughout the entire network. As each router receives the multicast traffic via its RPF interface (the interface in the direction of the source), it forwards the multicast traffic to all of its PIM-DM neighbors.

Note that this results in some traffic arriving via a non-RPF interface, as with the two routers in the center and far right of the figure. Packets arriving via the non-RPF interface are then discarded. These non-RPF flows are normal for the initial flooding of data and are corrected by the normal PIM-DM pruning mechanism.

Describing PIM-DM

This topic describes PIM-DM.



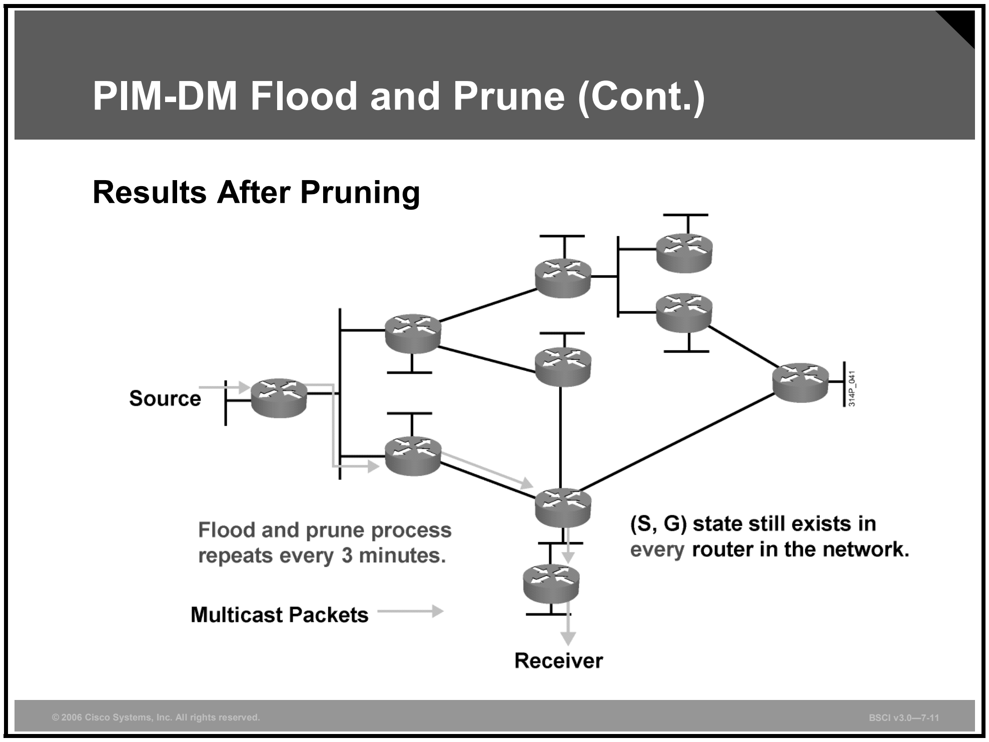
PIM-DM Flood and Prune (Cont.)

BSCI v3.0—7-10

In the example shown, PIM-DM prune messages are sent (denoted by dashed arrows) to stop the unwanted traffic.

Prune messages are also sent on non-RPF interfaces to shut off the flow of multicast traffic, because it is arriving via an interface that is not on the shortest path to the source. The example of prune messages sent on a non-RPF interface may be seen on the routers in the middle and far right of the figure.

Prune messages are sent on an RPF interface only when the router has no downstream receivers for multicast traffic from the specific source.



PIM-DM Flood and Prune (Cont.)

Results After Pruning

Flood and prune process repeats every 3 minutes.

(S, G) state still exists in every router in the network.

Multicast Packets

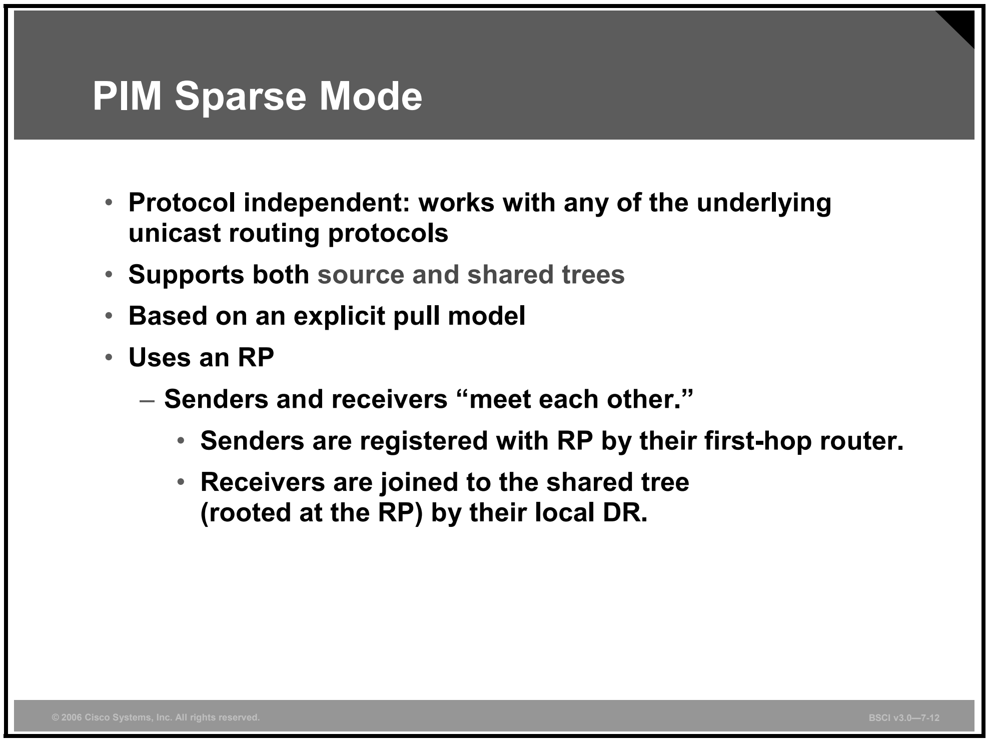
The example shows the SPT resulting from pruning the unwanted multicast traffic in the network.

Although the flow of multicast traffic is no longer reaching most of the routers in the network, the (S, G) state remains for all of them and will remain until the source stops sending.

In PIM-DM, all prune messages expire in 3 minutes. After that, the multicast traffic is flooded again to all of the routers. This periodic flood-and-prune behavior is normal and must be taken into account when the network is designed to use PIM-DM.

Describing PIM-SM

This topic describes PIM sparse mode (PIM-SM).



PIM Sparse Mode

- Protocol independent: works with any of the underlying unicast routing protocols
- Supports both source and shared trees
- Based on an explicit pull model
- Uses an RP
  - Senders and receivers "meet each other."
    - Senders are registered with RP by their first-hop router.
    - Receivers are joined to the shared tree (rooted at the RP) by their local DR.

BSCI v3.0—7-12

PIM-SM is described in RFC 2362. As with PIM-DM, PIM-SM is also independent of underlying unicast protocols. PIM-SM uses shared distribution trees, but it may also switch to the SPT.

PIM-SM is based on an explicit pull model. Therefore, traffic is forwarded only to the parts of the network that need it.

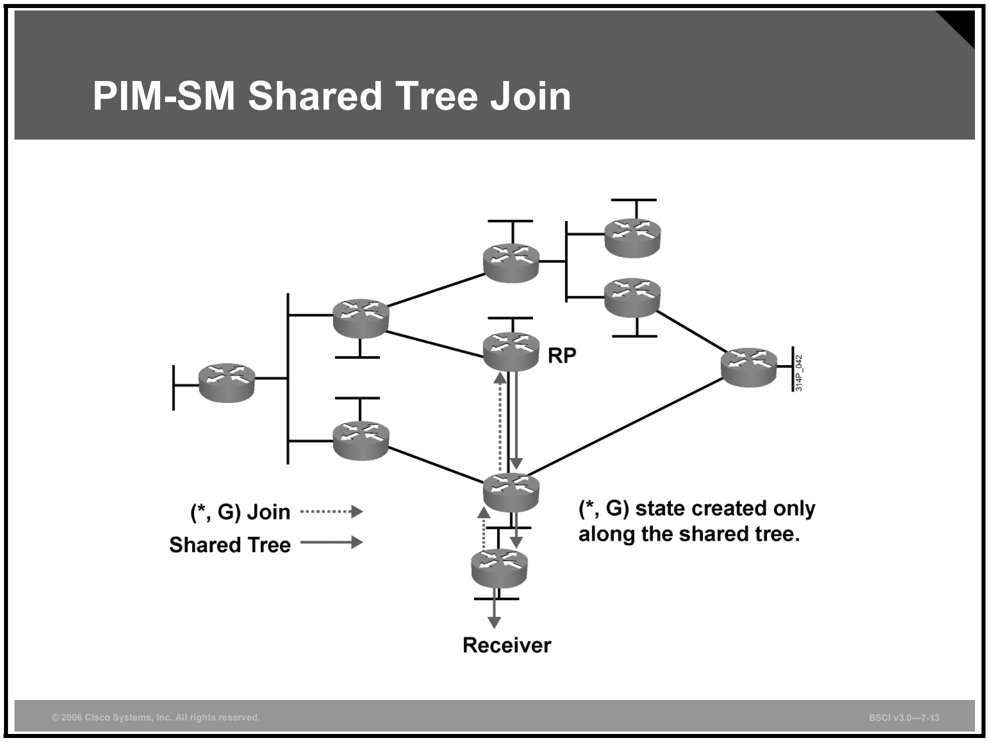
PIM-SM uses an RP to coordinate forwarding of multicast traffic from a source to receivers. Senders register with the RP and send a single copy of multicast data through it to the registered receivers.

Group members are joined to the shared tree by their local designated router (DR). A shared tree that is built this way is always rooted at the RP.

PIM-SM is appropriate for wide-scale deployment for both densely and sparsely populated groups in the enterprise network. It is the optimal choice for all production networks regardless of size and membership density.

There are many optimizations and enhancements to PIM, including the following:

- Bidirectional PIM mode, which is designed for many-to-many applications (that is, many hosts multicasting to each other).
- Source Specific Multicast (SSM) is a variant of PIM-SM that builds only source-specific SPTs and does not need an active RP for source-specific groups (address range 232/8).



PIM-SM Shared Tree Join

(*, G) state created only along the shared tree.

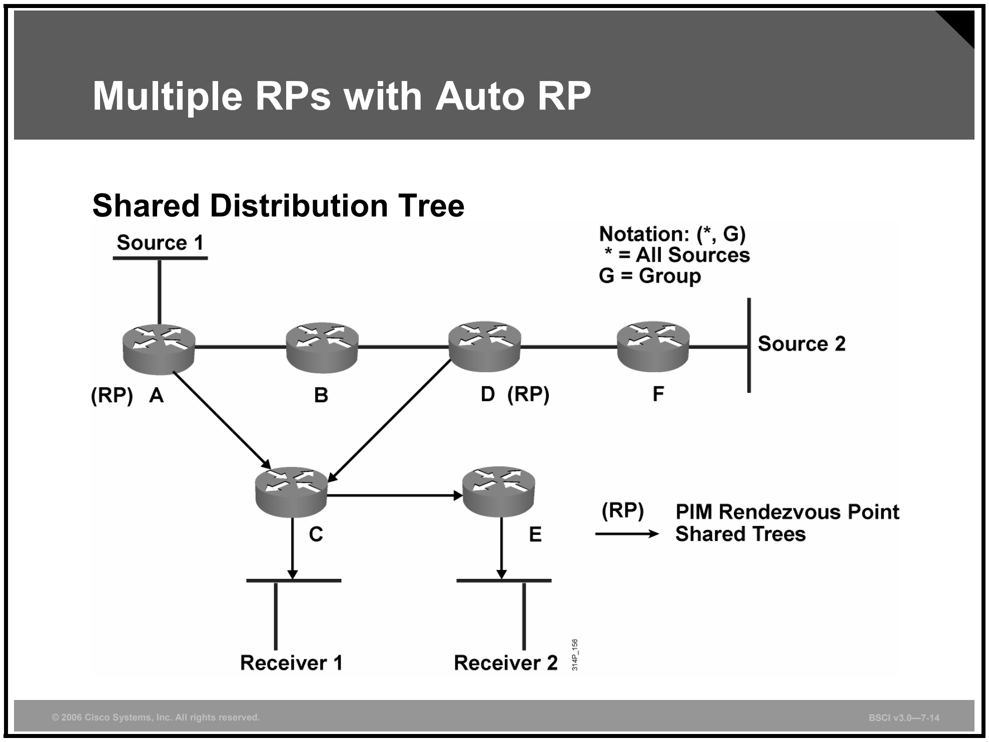
BSCI v3.0—7-13

In the example in the figure, an active receiver (attached to a leaf router at the bottom of the figure) has joined multicast group G.

The last-hop router knows the IP address of the RP router for group G, and it sends a (*, G) join for this group toward the RP.

This (*, G) join travels hop-by-hop toward the RP, building a branch of the shared tree that extends from the RP to the last-hop router directly connected to the receiver.

At this point, group G traffic may flow down the shared tree to the receiver.



Multiple RPs with Auto RP

Shared Distribution Tree

Notation: (*, G)
 * = All Sources
G = Group

(RP) PIM Rendezvous Point
→ Shared Trees

BSCI v3.0—7-14

The figure depicts two multicast sources. For maximum efficiency, multiple RPs can be implemented, with each RP in an optimum location. This design is difficult to configure, manage, and troubleshoot with manual configurations of RPs.

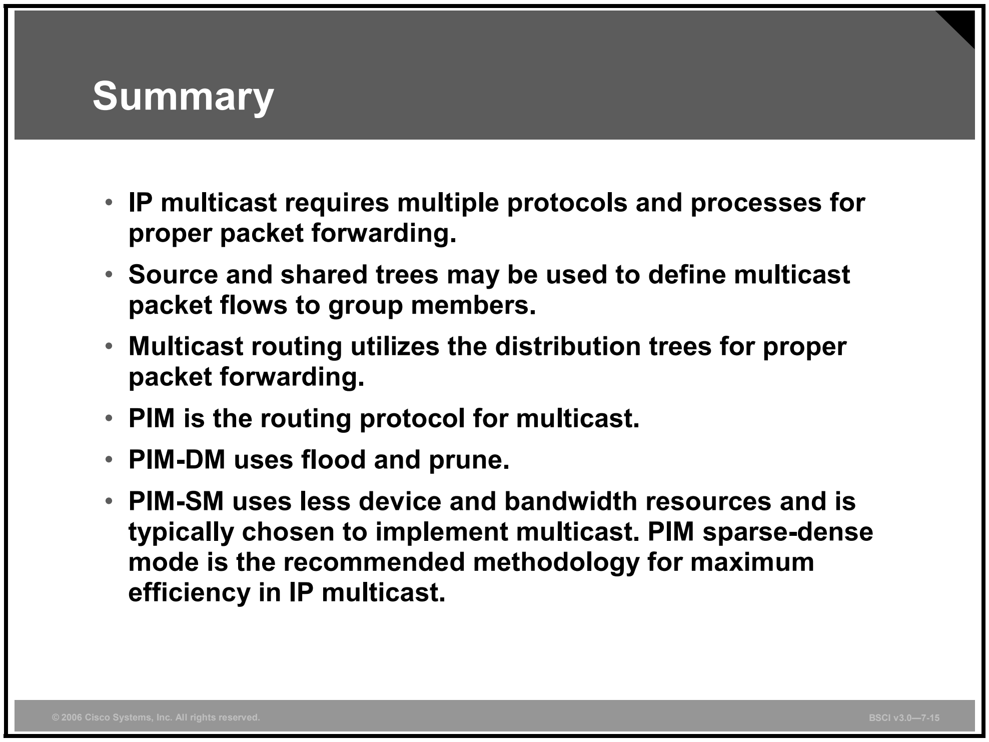
PIM sparse-dense mode supports automatic selection of RPs for each multicast. Router A in the figure could be the RP for source 1, and router F could be the RP for source 2.

PIM sparse-dense mode is the recommended solution from Cisco for IP multicast, because PIM-DM does not scale well and requires heavy router resources and PIM-SM offers limited RP configuration options.

If no RP is discovered for the multicast group or none is manually configured, PIM sparse-dense mode operates in dense mode. Therefore, automatic RP discovery should be implemented with PIM sparse-dense mode.

Summary

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



Summary

- **IP multicast requires multiple protocols and processes for proper packet forwarding.**
- **Source and shared trees may be used to define multicast packet flows to group members.**
- **Multicast routing utilizes the distribution trees for proper packet forwarding.**
- **PIM is the routing protocol for multicast.**
- **PIM-DM uses flood and prune.**
- **PIM-SM uses less device and bandwidth resources and is typically chosen to implement multicast. PIM sparse-dense mode is the recommended methodology for maximum efficiency in IP multicast.**

Lesson 4

Multicast Configuration and Verification

Overview

This lesson covers configuration, verification, troubleshooting, and related commands to implement Protocol Independent Multicast (PIM) sparse-dense mode and verify Internet Group Management Protocol (IGMP) snooping. It provides technical implementation details to support previous lessons.

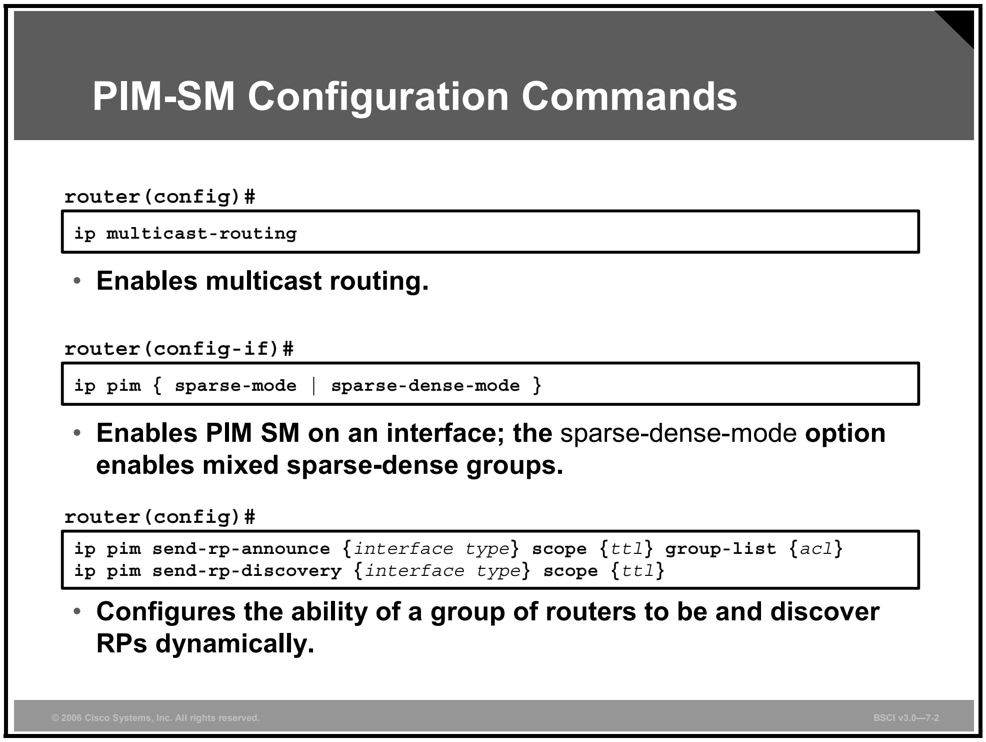
Objectives

Upon completing this lesson, you will be able to configure, verify, and test PIM sparse-dense mode and verify IGMP snooping. This ability includes being able to meet these objectives:

- Explain how to enable PIM sparse-dense mode on an interface
- Describe how to verify IGMP snooping

Enabling PIM-SM and PIM Sparse-Dense Mode on an Interface

This topic explains how to enable PIM sparse mode (PIM-SM) and PIM sparse-dense mode on an interface.



PIM-SM Configuration Commands

router(config)#

```
ip multicast-routing
```

- **Enables multicast routing.**

router(config-if)#

```
ip pim { sparse-mode | sparse-dense-mode }
```

- **Enables PIM SM on an interface; the** sparse-dense-mode **option enables mixed sparse-dense groups.**

router(config)#

```
ip pim send-rp-announce {interface type} scope {ttl} group-list {acl}
ip pim send-rp-discovery {interface type} scope {ttl}
```

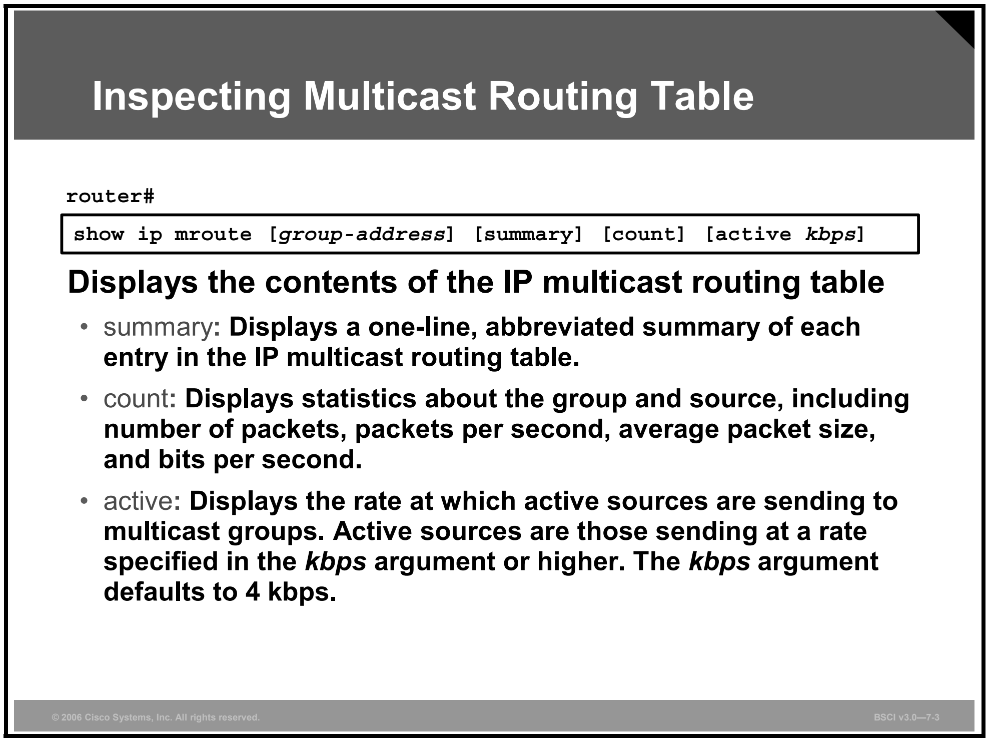
- **Configures the ability of a group of routers to be and discover RPs dynamically.**

BSCI v3.0—7-2

The commands needed for simple PIM-SM and PIM sparse-dense mode deployment are the following:

- The global command **ip multicast-routing** enables support for IP multicast on a router.
- The interface command **ip pim sparse-mode** enables PIM-SM operation on the selected interface. The **ip pim sparse-dense-mode** command enables the interface on the router to operate in PIM-SM for sparse-mode groups (those with known rendezvous points [RPs]) and in dense mode for other groups.
- The global command **ip pim send-rp-announce** {*interface type*} **scope** {*ttl*} **group-list** {**acl**} is issued on the router that you want to be an RP. This router sends an Auto-RP message to 224.0.1.39, announcing the router as a candidate RP for the groups in the range described by the access list.
- The global command **ip pim send-rp-discovery** {*interface type*} **scope** {*ttl*} configures the router as an RP mapping agent; it listens to the 224.0.1.39 address and sends a RP-to-group mapping message to 224.0.1.40. Other PIM routers listen to 224.0.1.40 to automatically discover the RP.
- The **ip pim spt-threshold** {*rate* | *infinity*} command controls the switchover from the shared distribution tree to the shortest path tree (SPT, or source distribution tree) in sparse mode. The keyword **infinity** means the switchover will never occur.

Note The recommended method for configuring an interface for PIM-SM operation is to use the **ip pim sparse-dense-mode** interface command. This method permits Auto-RP, bootstrap router (BSR), or statically defined RPs to be used with the least configuration effort.



Inspecting Multicast Routing Table

```
router#
show ip mroute [group-address] [summary] [count] [active kbps]
```

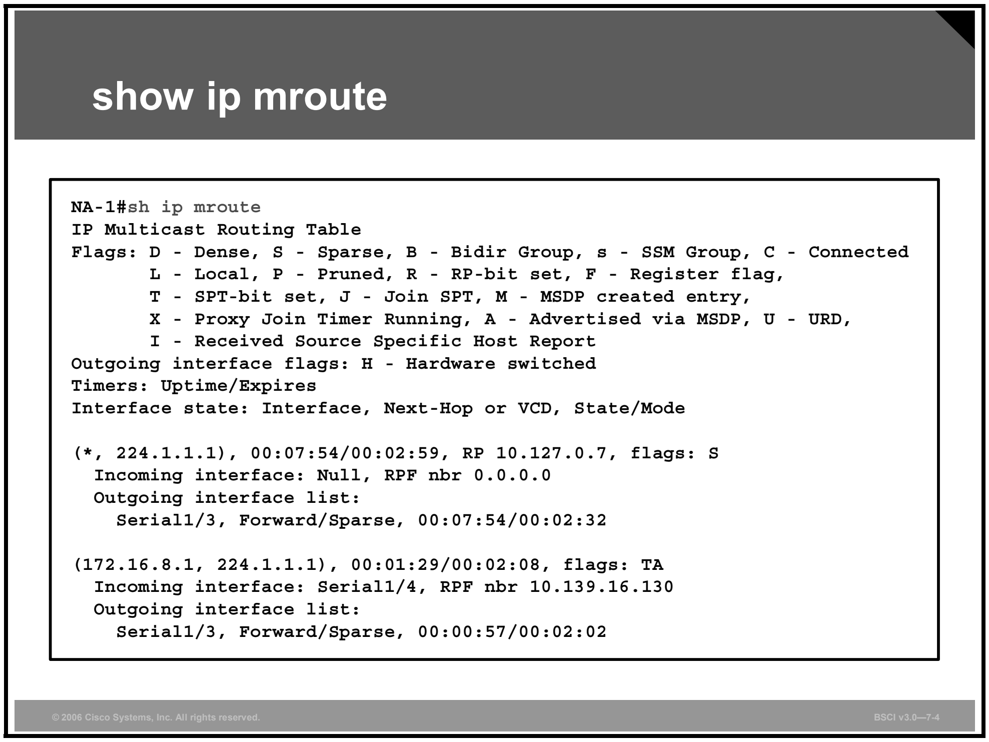
Displays the contents of the IP multicast routing table

- summary: **Displays a one-line, abbreviated summary of each entry in the IP multicast routing table.**
- count: **Displays statistics about the group and source, including number of packets, packets per second, average packet size, and bits per second.**
- active: **Displays the rate at which active sources are sending to multicast groups. Active sources are those sending at a rate specified in the *kbps* argument or higher. The *kbps* argument defaults to 4 kbps.**

The **show ip mroute** command is the most useful command for determining the state of multicast sources and groups, as recognized from the selected router perspective.

The output of the command generally represents a part of the multicast distribution tree, with an incoming interface and a list of outgoing interfaces. The following options may be used:

- **Summary:** Displays a one-line, abbreviated summary of each entry in the IP multicast routing table.
- **Count:** Displays statistics about the group and source, including number of packets, packets per second, average packet size, and bits per second.
- **Active:** Displays the rate at which active sources are sending to multicast groups. Active sources are those sending at a rate specified in the *kbps* argument or higher. The *kbps* argument defaults to 4 kbps.

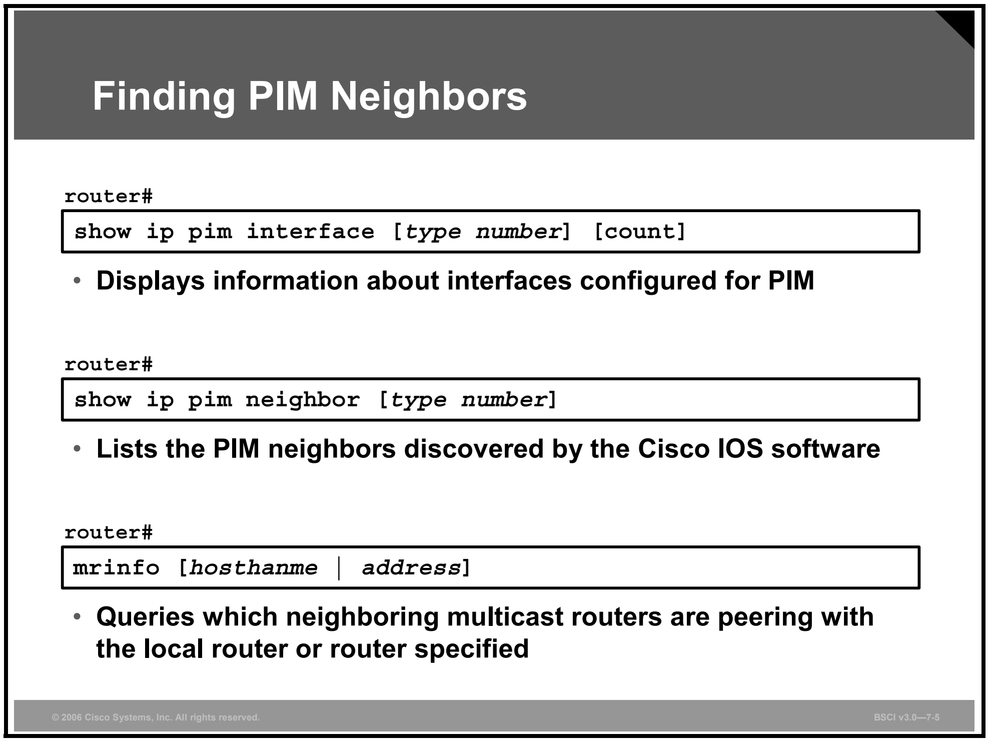


show ip mroute

```
NA-1#sh ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected
       L - Local, P - Pruned, R - RP-bit set, F - Register flag,
       T - SPT-bit set, J - Join SPT, M - MSDP created entry,
       X - Proxy Join Timer Running, A - Advertised via MSDP, U - URD,
       I - Received Source Specific Host Report
Outgoing interface flags: H - Hardware switched
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:07:54/00:02:59, RP 10.127.0.7, flags: S
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial1/3, Forward/Sparse, 00:07:54/00:02:32

(172.16.8.1, 224.1.1.1), 00:01:29/00:02:08, flags: TA
  Incoming interface: Serial1/4, RPF nbr 10.139.16.130
  Outgoing interface list:
    Serial1/3, Forward/Sparse, 00:00:57/00:02:02
```

The output of the **show ip mroute** command in the example shows a multicast routing table in PIM-SM environment.

- **(*, G) entry:** Timers, the RP address for the group, and the flags for the group (S is "sparse") are listed.
  - — The incoming interface is the interface toward the RP—if it is "Null," the router itself is the RP. The Reverse Path Forwarding (RPF) neighbor is the next-hop address toward the RP—if it is "0.0.0.0," the router is the RP for the group.
  - — The outgoing interface list (OIL) is a list of outgoing interfaces, along with modes and timers.
- **(S, G) entry:** Timers and flags for the entry are listed ("T" indicates that it is on the SPT; "A" indicates that it is to be advertised by Multicast Source Discovery Protocol [MSDP]).
  - — The incoming interface is the interface toward the source S. The RPF neighbor is the next-hop address toward the source—if it is "0.0.0.0," the source is directly attached.
  - — The OIL is a list of outgoing interfaces, in addition to modes and timers.



Finding PIM Neighbors

```
router#
show ip pim interface [type number] [count]
```

- **Displays information about interfaces configured for PIM**

```
router#
show ip pim neighbor [type number]
```

- **Lists the PIM neighbors discovered by the Cisco IOS software**

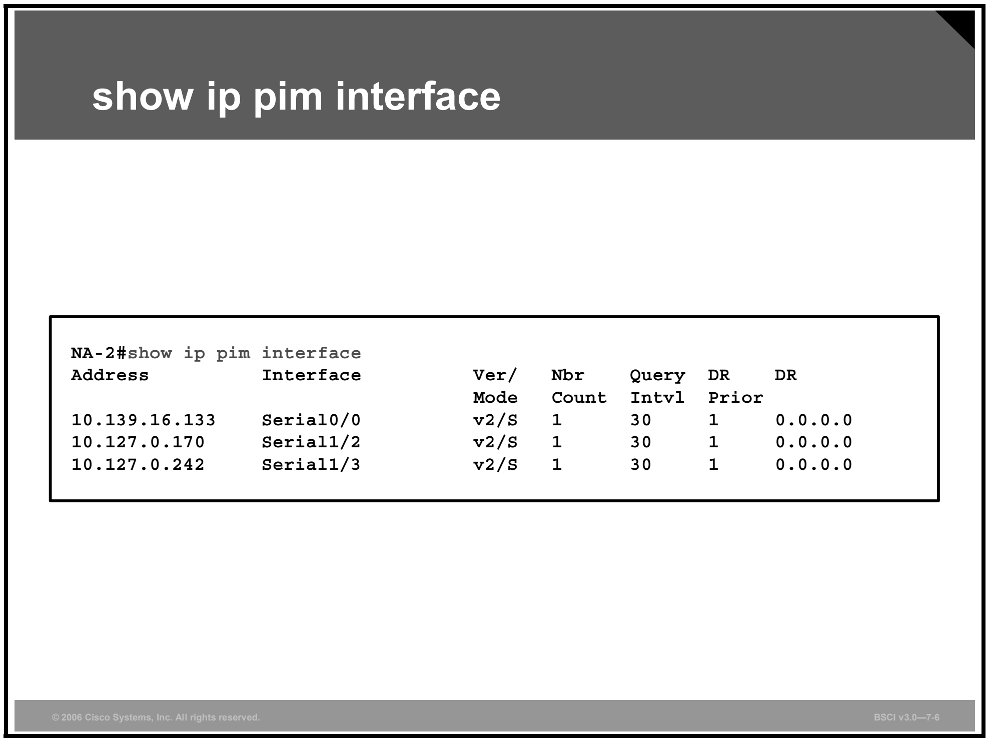
```
router#
mrinfo [hosthanme | address]
```

- **Queries which neighboring multicast routers are peering with the local router or router specified**

BSCI v3.0—7-5

When PIM-SM is configured, the first step in checking proper operation is to check PIM-enabled interfaces and to determine whether the PIM neighbors are correct. The following commands may be used to accomplish this:

- **show ip pim interface:** Displays the information about interfaces configured for PIM
- **show ip pim neighbor:** Displays the discovered PIM neighbors
- **mrinfo:** Displays information on multicast routers that are peering with the local router (no address) or with the addressed router

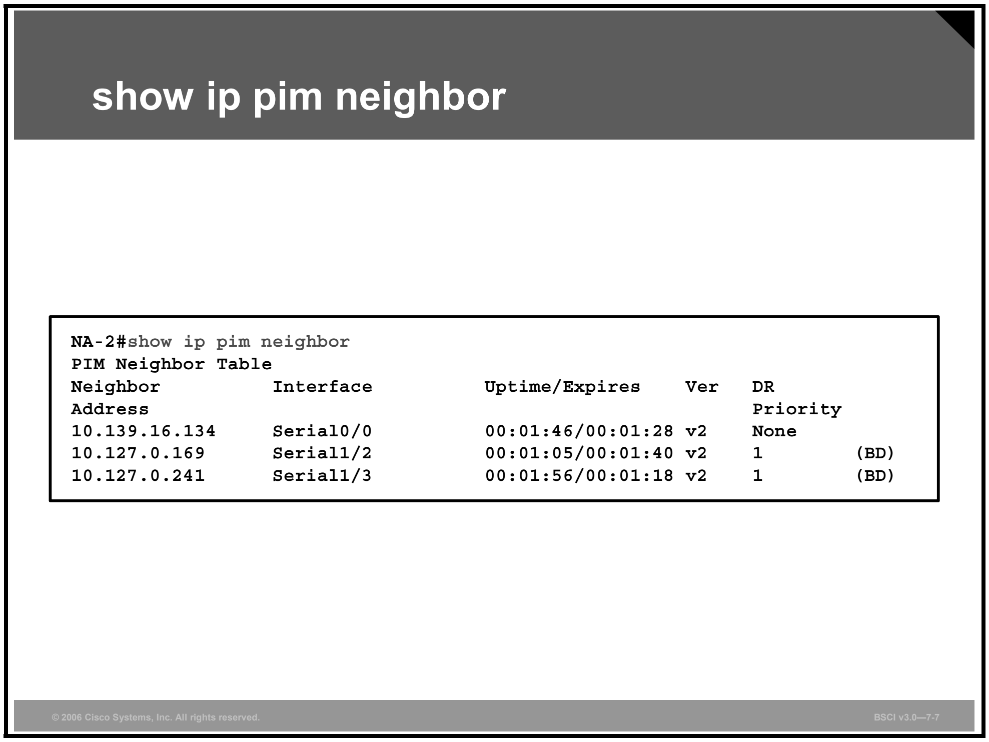


show ip pim interface

```
NA-2#show ip pim interface
Address          Interface          Ver/   Nbr    Query  DR     DR
                                    Mode   Count  Intvl  Prior
10.139.16.133    Serial0/0          v2/S   1      30     1      0.0.0.0
10.127.0.170     Serial1/2          v2/S   1      30     1      0.0.0.0
10.127.0.242     Serial1/3          v2/S   1      30     1      0.0.0.0
```

The **show ip pim interface** command output contains the following information:

- **Address:** IP address of the interface.
- **Interface:** Type and number of the interface configured for PIM.
- **Ver/mode:** PIM version (1 or 2) that is running on the interface and the mode (dense mode, sparse mode, or sparse-dense mode).
- **Nbr Count:** Number of neighbors on this link.
- **Query Intvl:** Frequency at which PIM hellos and PIM queries are sent (default is 30 seconds).
- **DR Prior:** Priority used in designated router (DR) election. If all the routers on a multiaccess link have the same priority (default = 1), the highest IP address is a tiebreaker.
- **DR:** IP address of the designated router. (On point-to-point links, there are no DRs, thus the output shows 0.0.0.0).



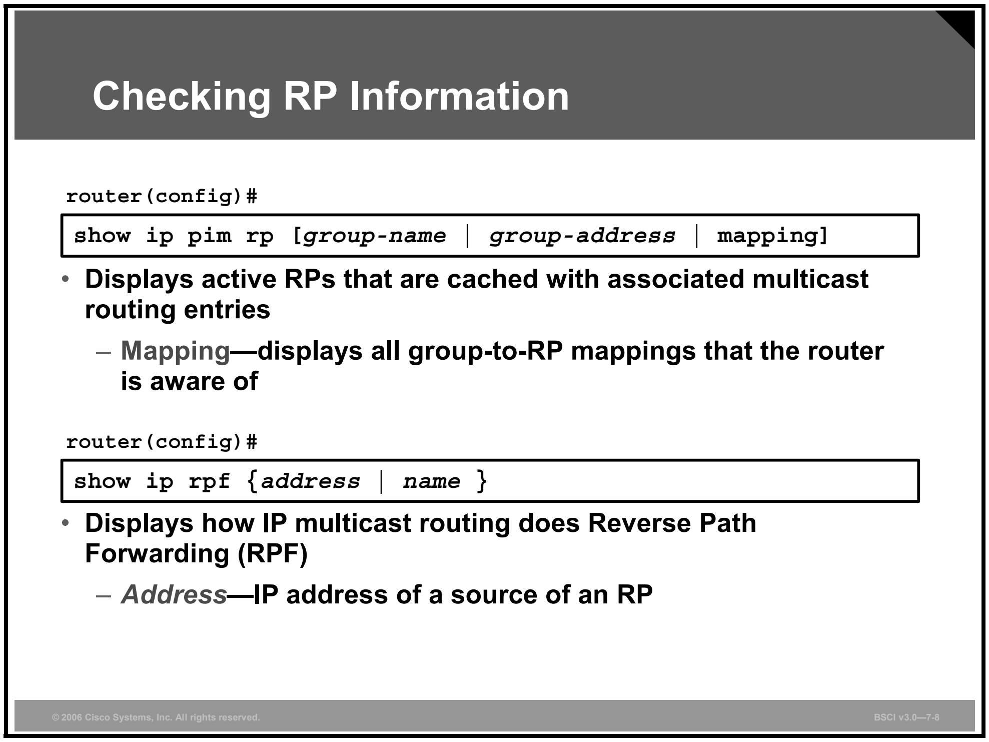
show ip pim neighbor

```
NA-2#show ip pim neighbor
PIM Neighbor Table
Neighbor          Interface          Uptime/Expires    Ver   DR
Address                                                      Priority
10.139.16.134     Serial0/0          00:01:46/00:01:28 v2    None
10.127.0.169      Serial1/2          00:01:05/00:01:40 v2    1        (BD)
10.127.0.241      Serial1/3          00:01:56/00:01:18 v2    1        (BD)
```

BSCI v3.0—7-7

The **show ip pim neighbor** command displays the following information:

- **Neighbor Address:** IP address of the PIM neighbor.
- **Interface:** Interface where the PIM hello (PIM query in PIMv1) of this neighbor was received.
- **Uptime:** Period of time that this PIM neighbor has been active.
- **Expires:** Period of time (hold time) after which this PIM neighbor will no longer be considered active. Receipt of another PIM hello or PIM query resets the timer.
- **Ver:** PIM version that the neighbor is using (version 1 or 2).
- **DR Priority:** If the neighbor supports this option, the numeric value is shown; if no number is shown, the option is not supported by the neighbor.



Checking RP Information

```
router(config)#
```

```
show ip pim rp [group-name | group-address | mapping]
```

- **Displays active RPs that are cached with associated multicast routing entries**
  - Mapping**—displays all group-to-RP mappings that the router is aware of**

```
router(config)#
```

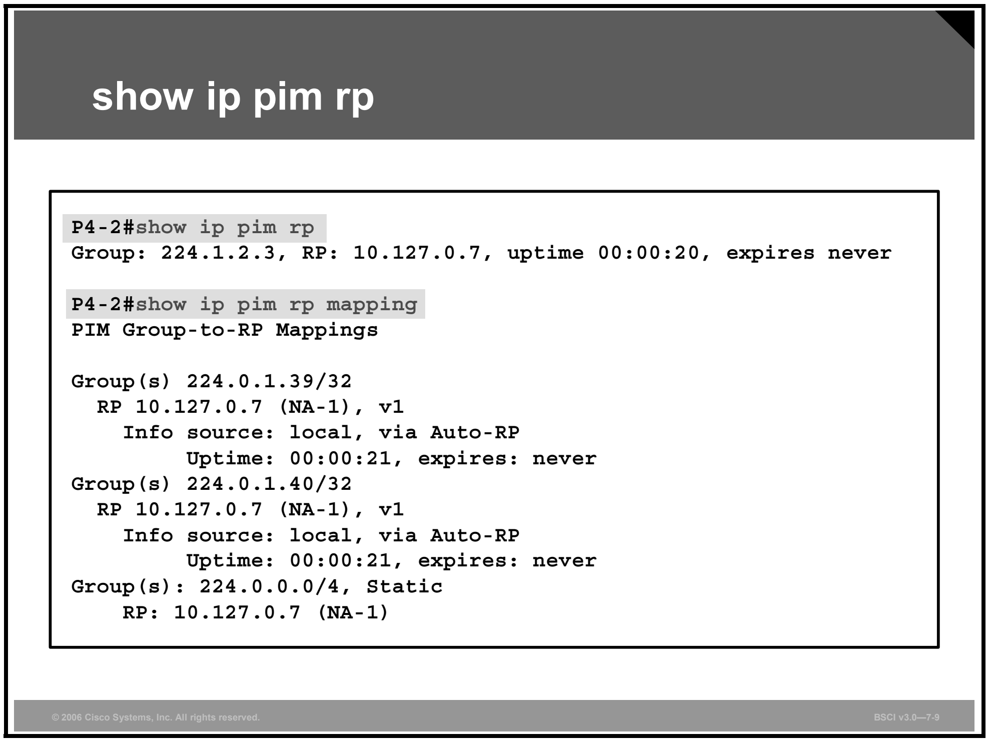
```
show ip rpf {address | name }
```

- **Displays how IP multicast routing does Reverse Path Forwarding (RPF)**
  - *Address***—IP address of a source of an RP**

BSCI v3.0—7-8

The RP for a certain multicast group operating in PIM-SM has to be reachable and known to the router. Troubleshooting RP information requires, in addition to standard tools used in unicast ping to check RP reachability, at least the following three commands:

- **show ip pim rp:** Displays, without arguments, RP information on active groups. If the group address or name is provided, only the RP information for the selected group is shown (assuming that it is an active group).
- **show ip pim rp mapping:** Displays the contents of the important group-to-RP mapping cache that contains the information about which RP is active for which group range. The group-to-RP mapping cache is populated by the Auto-RP or BSR mechanisms and by static RP assignments. It is very important to check this information to verify that the router possesses the RP mapping information consistent with proper network operation.
- **show ip rpf:** Displays RPF information for the RP or for the source.



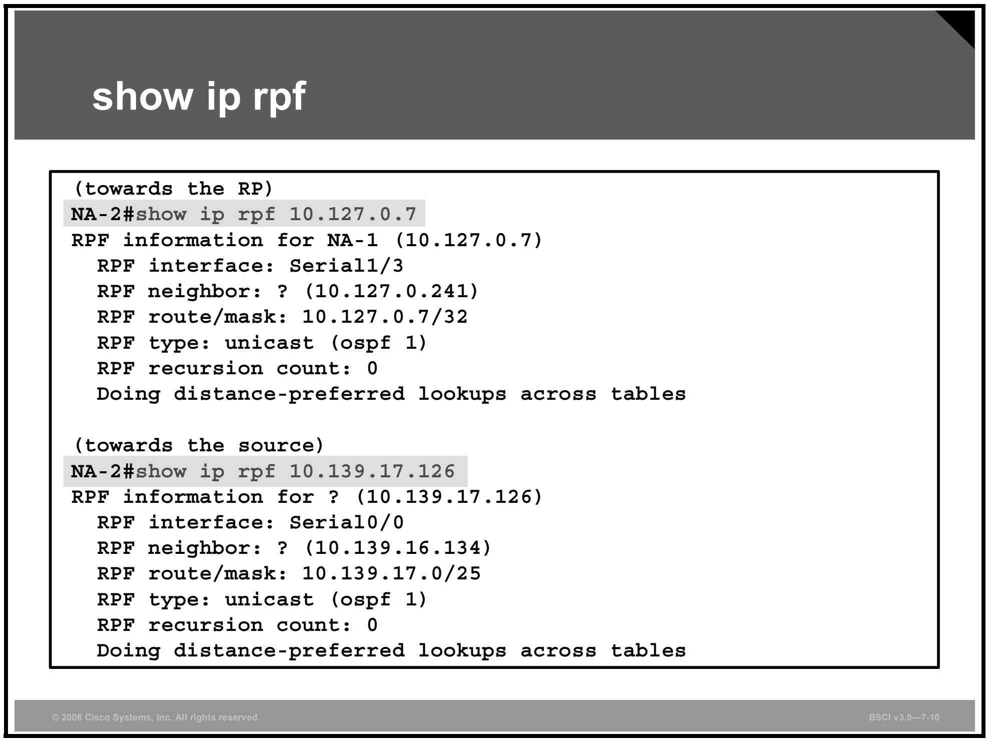
show ip pim rp

```
P4-2#show ip pim rp
Group: 224.1.2.3, RP: 10.127.0.7, uptime 00:00:20, expires never

P4-2#show ip pim rp mapping
PIM Group-to-RP Mappings

Group(s) 224.0.1.39/32
  RP 10.127.0.7 (NA-1), v1
    Info source: local, via Auto-RP
         Uptime: 00:00:21, expires: never
Group(s) 224.0.1.40/32
  RP 10.127.0.7 (NA-1), v1
    Info source: local, via Auto-RP
         Uptime: 00:00:21, expires: never
Group(s): 224.0.0.0/4, Static
    RP: 10.127.0.7 (NA-1)
```

The output of the **show ip pim rp** command simply lists all active groups and their associated RPs. This form of the command is becoming obsolete, because it offers limited information.

Instead, the **show ip pim rp mapping** command should be used in most cases, because it supplies details on the actual contents of the group-to-RP mapping cache, such as the following:

- IP address of a router that distributed the information or local—when the source of the information is a local router that either has manual RP configuration or is a source of automatically distributed information
- Mechanism by which this information was determined—Auto-RP, BSR, or static
- Whether or not this router is operating as a candidate-RP, mapping agent, or BSR (not shown in the figure)



show ip rpf

```
 (towards the RP)
NA-2#show ip rpf 10.127.0.7
RPF information for NA-1 (10.127.0.7)
  RPF interface: Serial1/3
  RPF neighbor: ? (10.127.0.241)
  RPF route/mask: 10.127.0.7/32
  RPF type: unicast (ospf 1)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables

 (towards the source)
NA-2#show ip rpf 10.139.17.126
RPF information for ? (10.139.17.126)
  RPF interface: Serial0/0
  RPF neighbor: ? (10.139.16.134)
  RPF route/mask: 10.139.17.0/25
  RPF type: unicast (ospf 1)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

BSCI v3.0—7-10

The output of the **show ip rpf** command displays RPF information associated with the specified source address. The specified address does not necessarily have to be a currently active source. In fact, it may be an IP address, including the address of the RP. Specifying the address of the RP is very useful in determining the RPF information for the shared tree.

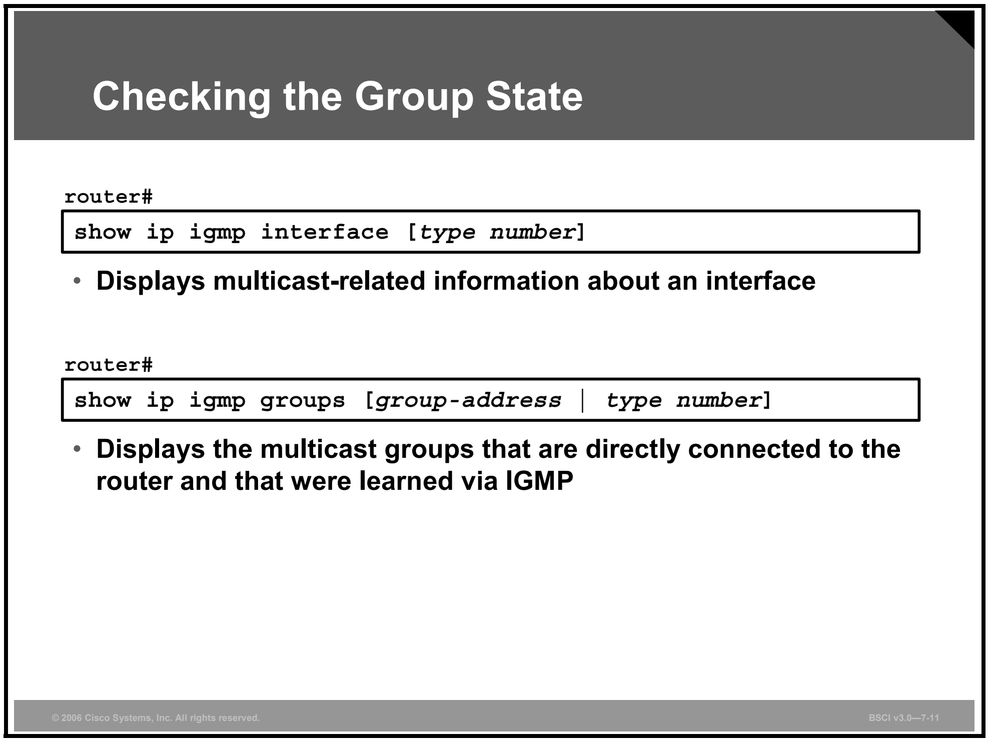
“RPF interface” is the interface in the direction of the source (or RP), while “RPF neighbor” is the address of the next-hop router in the direction of the source (or RP).

“RPF type” indicates the source of RPF information. In the example, “unicast” indicates that the information was derived from the unicast routing table (in this case, from Open Shortest Path First Protocol [OSPF]). Other RPF types include Distance Vector Multicast Routing Protocol (DVMRP), Multiprotocol Border Gateway Protocol (BGP) extensions for IP multicast, or static.

RPF information is essential in multicast routing, and special care has to be taken when inspecting the PIM-SM information because of the possible coexistence of shared trees and SPTs.

Verifying IGMP Groups and IGMP Snooping

This topic explains how to verify IGMP groups on a router and IGMP snooping on a switch.



Checking the Group State

```
router#
show ip igmp interface [type number]
```

- **Displays multicast-related information about an interface**

```
router#
show ip igmp groups [group-address | type number]
```

- **Displays the multicast groups that are directly connected to the router and that were learned via IGMP**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—7-11

If the multicast traffic is not flowing to receivers, the IGMP group membership has to be checked on the leaf routers. The **show ip igmp interface** command shows information about the selected interface, and **show ip igmp groups** lists the local groups known to the router.

Enabling PIM on an interface also enables IGMP operation on that interface. An interface can be configured to be in dense mode, sparse mode, or sparse-dense mode. The mode determines how the router populates its multicast routing table and how the router forwards multicast packets that it receives from its directly connected LANs. You must enable PIM in one of these modes for an interface to perform IP multicast routing.

Configure a Router to Be a Member of a Group or a Statically Connected Member

Sometimes either there is no group member on a network segment or a host cannot report its group membership using IGMP. However, you may want multicast traffic to go to that network segment. These commands are often used in lab environments where no multicast servers and receivers are configured. The following are two ways to pull multicast traffic down to a network segment:

- **ip igmp join-group:** With this command, the router accepts the multicast packets in addition to forwarding them. Accepting the multicast packets prevents the router from fast switching.

  Cisco Systems routers can be configured to be members of a multicast group, which is useful for determining multicast reachability in a network. If a device is configured to be a group member and supports the protocol that is being transmitted to the group, it can respond (for example, to the **ping** command). The device responds to Internet Control Message Protocol (ICMP) echo request packets addressed to a group of which it is a member.

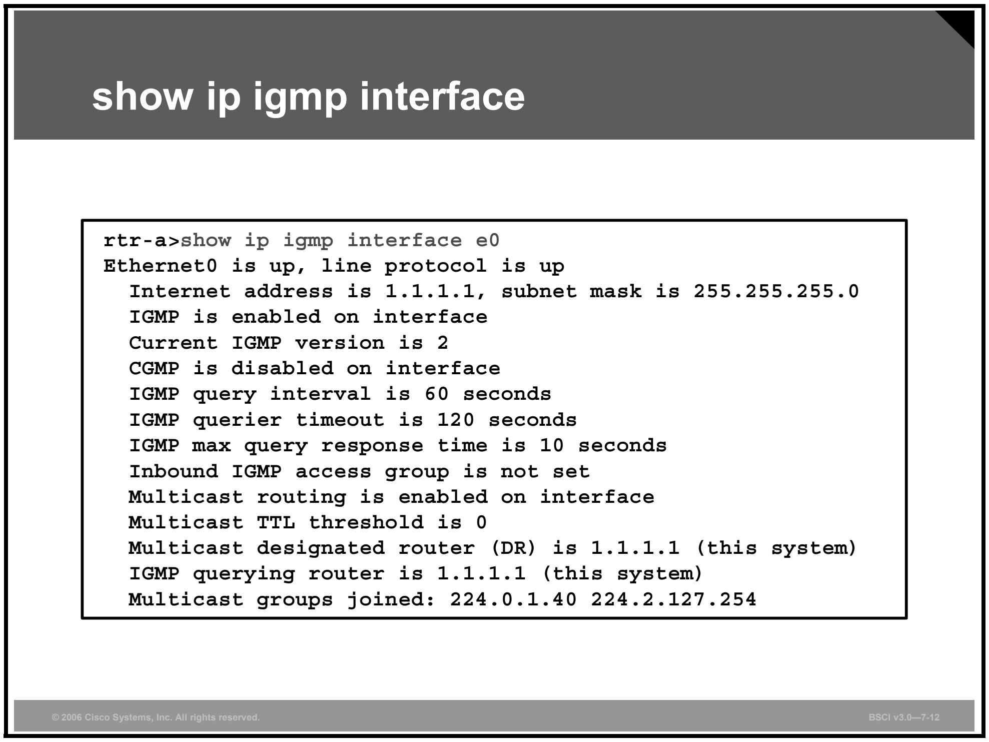
Another example is the multicast traceroute tools provided in the Cisco IOS software. To have the router join a multicast group and enable IGMP, use the command shown in the table in interface configuration mode. This command is often used in lab environments where no multicast servers and receivers are configured.

Command	Purpose
ip igmp join-group *group-address*	Joins a multicast group

- **ip igmp static-group:** With this command, the router does not accept the packets itself but only forwards them. Hence, this method allows fast switching. The outgoing interface appears in the IGMP cache, but the router itself is not a member, as evidenced by lack of an "L" (local) flag in the multicast route entry.

To configure the router itself to be a statically connected member of a group (and allow fast switching), use the command shown in the table in interface configuration mode.

Command	Purpose
ip igmp static-group *group-address*	Configures the router as a statically connected member of a group



show ip igmp interface

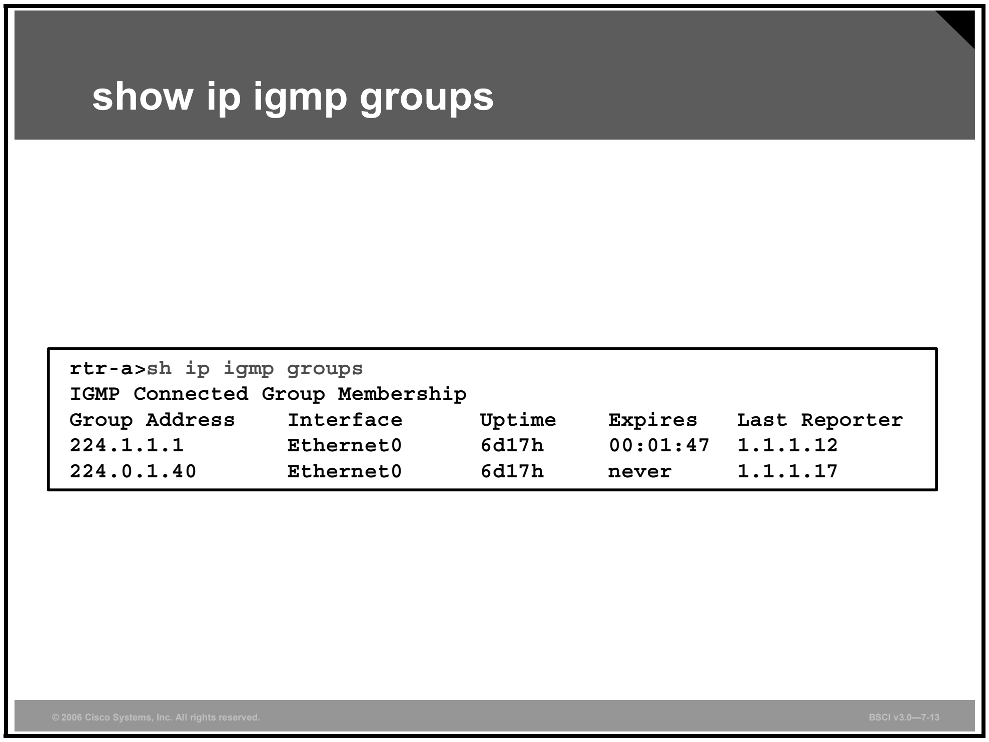
```
rtr-a>show ip igmp interface e0
Ethernet0 is up, line protocol is up
  Internet address is 1.1.1.1, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Inbound IGMP access group is not set
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 1.1.1.1 (this system)
  IGMP querying router is 1.1.1.1 (this system)
  Multicast groups joined: 224.0.1.40 224.2.127.254
```

Use the **show ip igmp groups** command to display the multicast groups that are directly connected to the router and that were learned via IGMP. This command is used to determine the following information:

- Interface configuration for multicast and IGMP
- Version for which the IGMP interface is configured
- IGMPv2 querier on the multiaccess network
- Multicast designated router (DR)
- Joined multicast groups on the current router

In the example in the figure, the router itself joins these two groups:

- **224.0.1.40 group:** Auto-RP, which is joined automatically
- **224.2.127.254 group:** Session Description Protocol tool (SDR), which was joined by configuring the **ip sdr listen** command on the interface



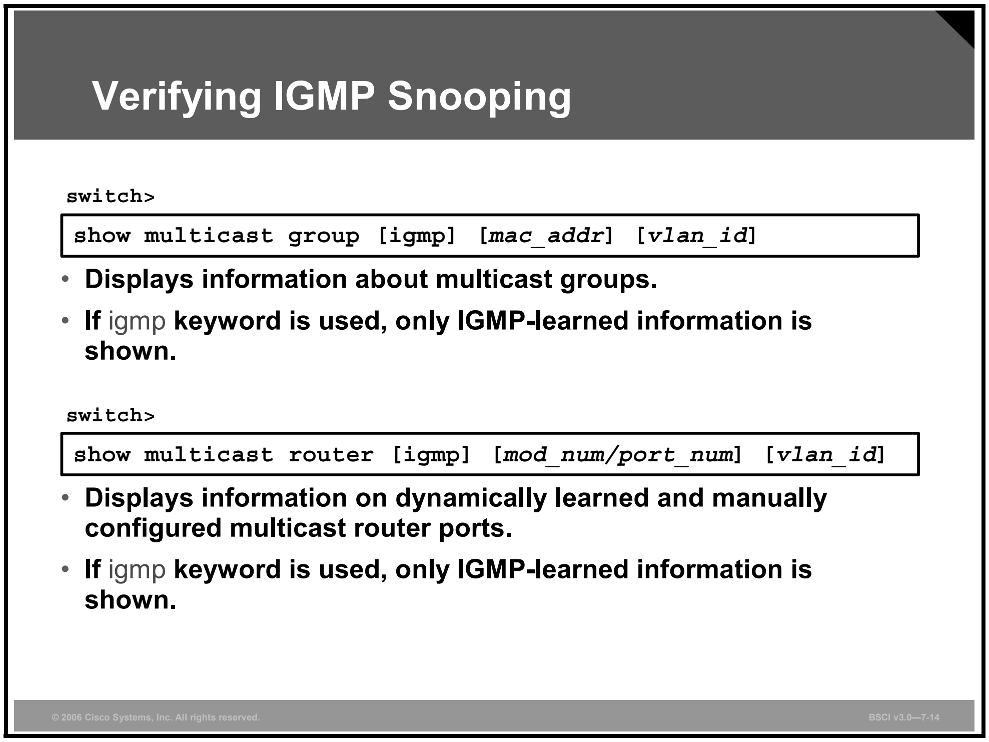
show ip igmp groups

```
rtr-a>sh ip igmp groups
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0      6d17h     00:01:47  1.1.1.12
224.0.1.40       Ethernet0      6d17h     never     1.1.1.17
```

BSCI v3.0—7-13

In the example, the router recognizes these two multicast groups:

- Group 224.1.1.1 is active on Ethernet0 and has been active on this interface for 6 days and 17 hours. This group expires (and will be deleted) in 1 minute and 47 seconds if an IGMP host membership report for this group is not heard in that time. The last host to report membership was 1.1.1.12.
- Group 224.0.1.40 (Auto-RP) is automatically joined by all Cisco routers. Thus, its expiration shows as “never.”



Verifying IGMP Snooping

switch>

```
show multicast group [igmp] [mac_addr] [vlan_id]
```

- **Displays information about multicast groups.**
- **If** igmp **keyword is used, only IGMP-learned information is shown.**

switch>

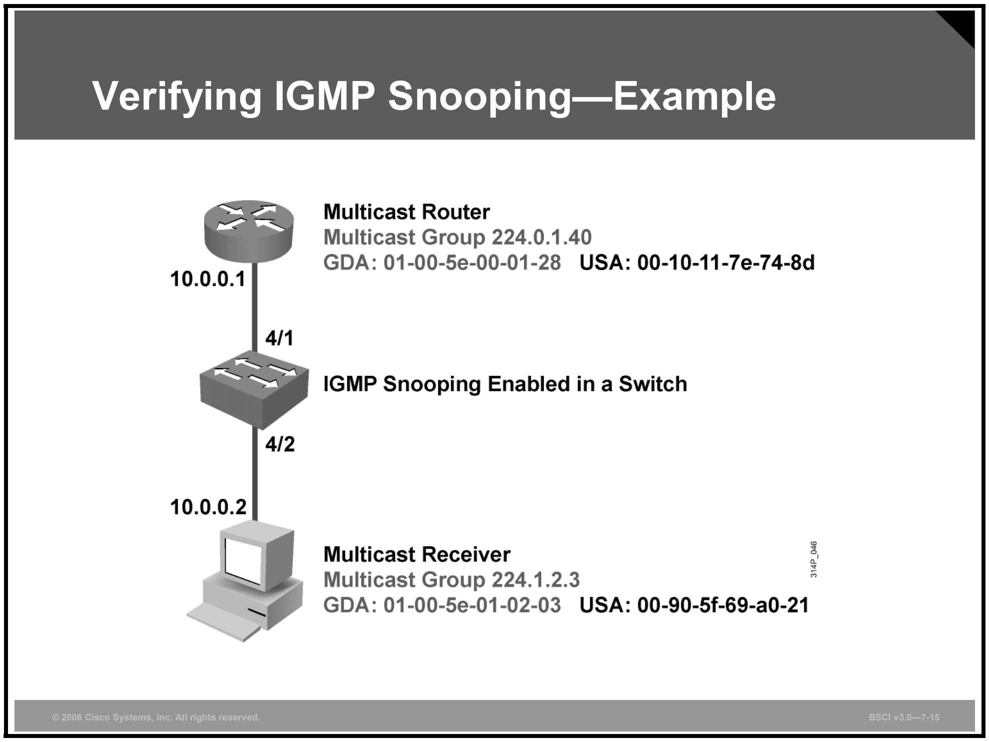
```
show multicast router [igmp] [mod_num/port_num] [vlan_id]
```

- **Displays information on dynamically learned and manually configured multicast router ports.**
- **If** igmp **keyword is used, only IGMP-learned information is shown.**

BSCI v3.0—7-14

When verifying IGMP snooping on a switch, use the **show multicast group** command to display the multicast group configuration.

Use the **show multicast router** command to display the information about ports that have IGMP multicast routers connected to them. Use the **igmp** keyword to specify that only the configuration information learned through IGMP is to be displayed.



Verifying IGMP Snooping—Example

10.0.0.1

Multicast Router
Multicast Group 224.0.1.40
GDA: 01-00-5e-00-01-28 **USA: 00-10-11-7e-74-8d**

4/1

IGMP Snooping Enabled in a Switch

4/2

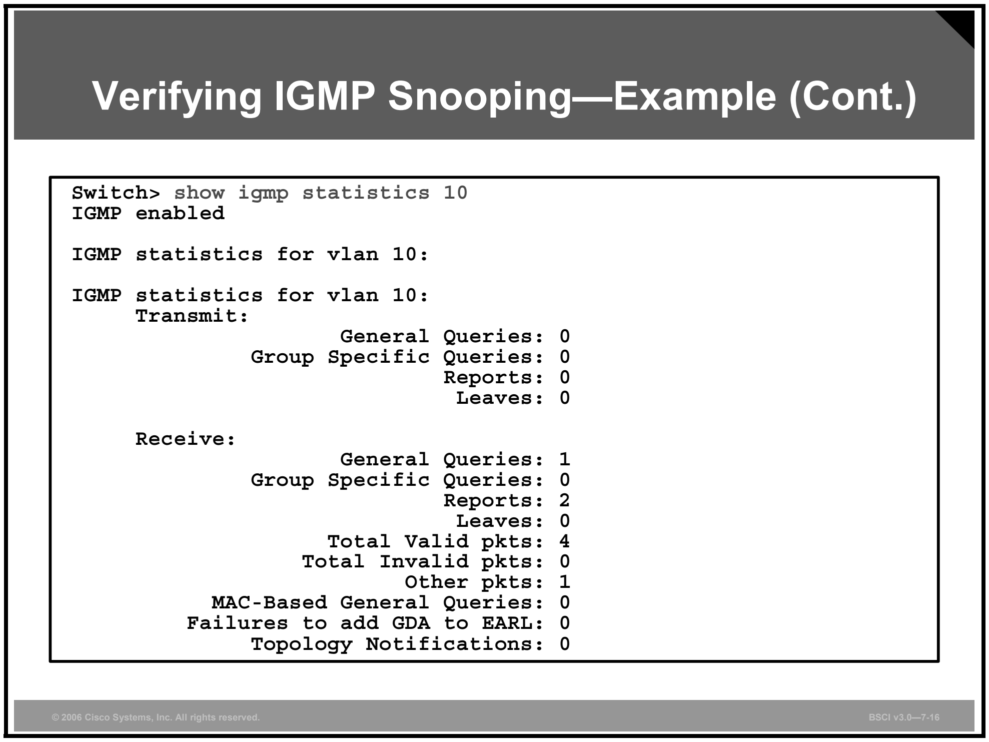
10.0.0.2

Multicast Receiver
Multicast Group 224.1.2.3
GDA: 01-00-5e-01-02-03 **USA: 00-90-5f-69-a0-21**

BSCI v3.0—7-15

The example shows an IGMP-enabled multicast router with the IP address 10.0.0.1 and a multicast receiver with the IP address 10.0.0.2 connected to a router via a switch that has IGMP snooping enabled over ports 4/1 and 4/2, respectively. Both ports are in VLAN 10.

- The IGMP router with the MAC address 00-10-11-7e-74-8d is a member of multicast group 224.0.1.40.
- A multicast receiver with the MAC address 00-90-5f-69-a0-21 is receiving multicast traffic from multicast group 224.1.2.3.

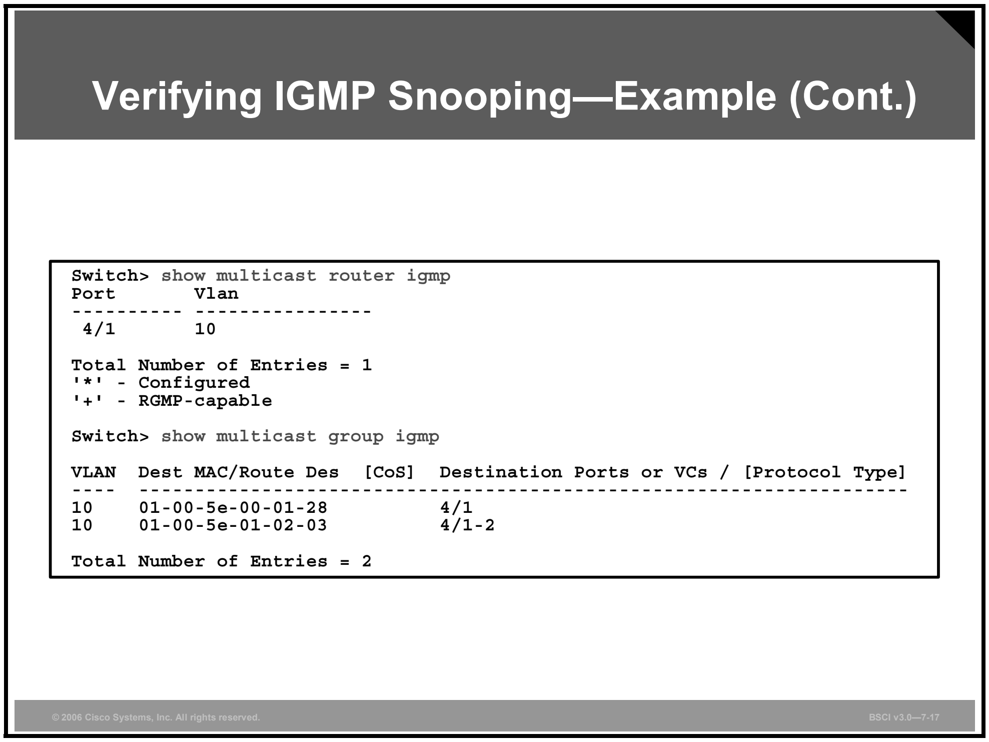


Verifying IGMP Snooping—Example (Cont.)

```
Switch> show igmp statistics 10
IGMP enabled

IGMP statistics for vlan 10:

IGMP statistics for vlan 10:
     Transmit:
                     General Queries: 0
              Group Specific Queries: 0
                             Reports: 0
                              Leaves: 0

     Receive:
                     General Queries: 1
              Group Specific Queries: 0
                             Reports: 2
                              Leaves: 0
                    Total Valid pkts: 4
                  Total Invalid pkts: 0
                          Other pkts: 1
            MAC-Based General Queries: 0
          Failures to add GDA to EARL: 0
               Topology Notifications: 0
```

The output of the **show igmp statistics** command displays sample statistics for VLAN 10. The number of the group-specific queries and general queries is shown, in addition to the number of host membership reports and group leave messages.

The statistics are split into receive and transmit information.



Verifying IGMP Snooping—Example (Cont.)

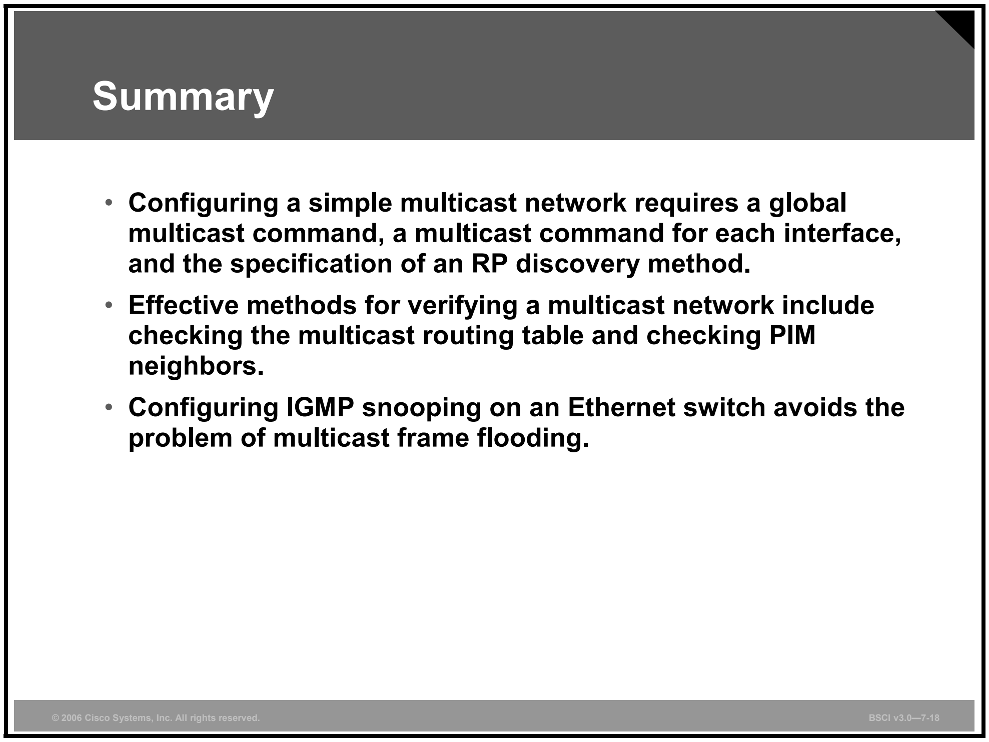
```
Switch> show multicast router igmp
Port       Vlan
---------- ----------------
 4/1       10

Total Number of Entries = 1
'*' - Configured
'+' - RGMP-capable

Switch> show multicast group igmp

VLAN  Dest MAC/Route Des  [CoS]  Destination Ports or VCs / [Protocol Type]
----  ---------------------------------------------------------------------
10    01-00-5e-00-01-28          4/1
10    01-00-5e-01-02-03          4/1-2

Total Number of Entries = 2
```

The **show multicast router igmp** command displays information about the IGMP router port and the VLAN configuration on a switch.

The **show multicast group igmp** command lists information about the VLAN, multicast group, and ports that are joined to that multicast group on a switch.

Summary

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

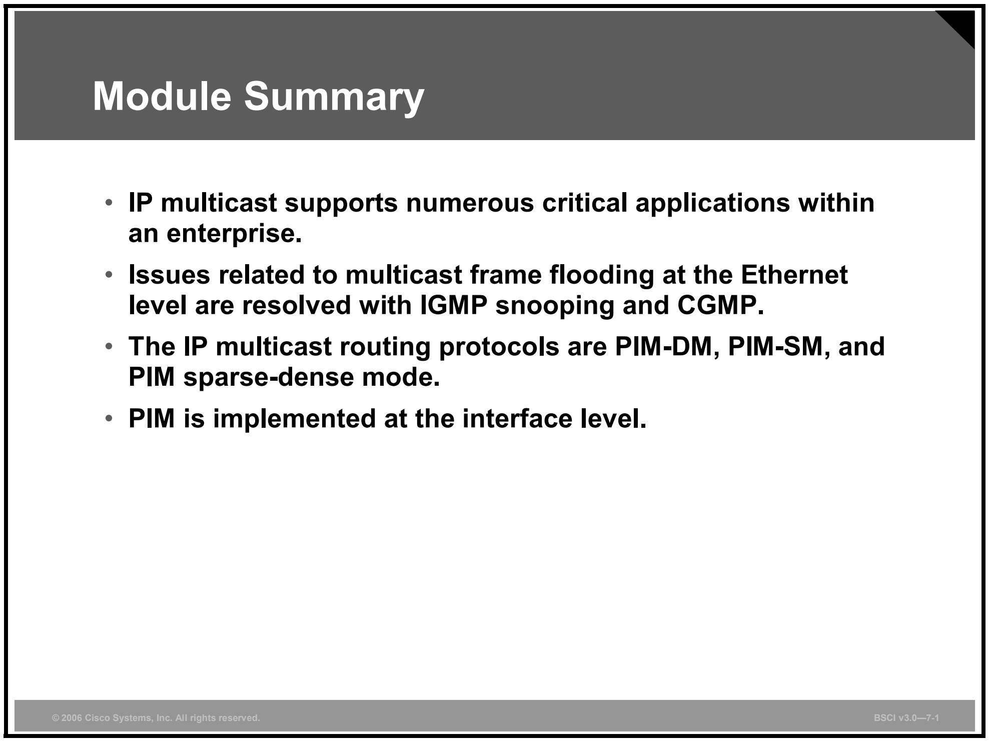


Summary

- **Configuring a simple multicast network requires a global multicast command, a multicast command for each interface, and the specification of an RP discovery method.**
- **Effective methods for verifying a multicast network include checking the multicast routing table and checking PIM neighbors.**
- **Configuring IGMP snooping on an Ethernet switch avoids the problem of multicast frame flooding.**

Module Summary

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



Module Summary

- **IP multicast supports numerous critical applications within an enterprise.**
- **Issues related to multicast frame flooding at the Ethernet level are resolved with IGMP snooping and CGMP.**
- **The IP multicast routing protocols are PIM-DM, PIM-SM, and PIM sparse-dense mode.**
- **PIM is implemented at the interface level.**

BSCI v3.0—7-1

This module covered IP multicast benefits and the types of applications supported. To achieve IP multicast forwarding, proper addresses for multicast groups must be assigned, then a tree is constructed from each source to the users that have joined the group. The tree is built using Reverse Path Forwarding (RPF). There are two type of trees, source and shared. The protocol to support multicast users joining a multicast group is Internet Group Management Protocol (IGMP).

A multicast routing protocol is required at Layer 3, and the choices are Protocol Independent Multicast dense mode (PIM-DM), PIM sparse mode (PIM-SM), or PIM sparse-dense mode. In PIM-SM, a shared distribution tree process is used and an RP is required. Layer 2 Ethernet switching issues are resolved with IGMP snooping or Cisco Group Management Protocol (CGMP).

Cisco IOS software provides a full set of configuration and verification commands for the implementation of IP multicast.

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 statement best describes unicast? (Source: Explaining Multicast)

A) There are multiple copies of data, one copy for each receiver.
B) Unicast sends a single copy of data to multiple receivers.
C) Unicast sends a copy of data to all users.
D) Unicast sends a copy of data to a well-known address.

Q2) Which of the following is NOT a disadvantage of multicast? (Source: Explaining Multicast)

A) no congestion avoidance
B) efficient use of network resources
C) possible duplication of packets
D) best-effort delivery
E) out-of-sequence delivery

Q3) Which two addresses would be considered limited-scope IP multicast addresses? (Choose two.) (Source: Explaining Multicast)

A) 239.46.35.46
B) 224.2.2.2
C) 229.13.13.13
D) 239.2.239.239

Q4) Which command shows the version of IGMP running on an interface? (Source: IGMP and Layer 2 Issues)

A) **show igmp**
B) **show interface**
C) **show ip igmp interface**
D) **show ip interface**

Q5) Which two solutions prevent flooding of multicast frames in an Ethernet switching environment? (Choose two.) (Source: IGMP and Layer 2 Issues)

A) VLAN access lists
B) PIM sparse-dense mode
C) IGMP snooping
D) CGMP

Q6) If a rendezvous point is configured, which two of the following statements are true? (Choose two.) (Source: Explaining Multicast Routing Protocols)

A) It is a shared tree.
B) The routing protocol is PIM-SM.
C) The routing protocol is PIM-DM.
D) It is a source tree.

Q7) Repeated flood and prune is a characteristic of which multicast routing protocol? (Source: Explaining Multicast Routing Protocols)

A) PIM-SM
B) PIM-DM
C) Multicast OSPF
D) PIM sparse-dense mode

Q8) Which is the global command to enable IP multicast routing? (Source: Multicast Configuration and Verification)

A) router(config)# **ip multicast-routing**
B) router(config)# **router pim**
C) router(config)# **mroute pim**
D) router(config)# **router mcast**

Q9) Which command provides information about PIM neighbors discovered by the router? (Source: Multicast Configuration and Verification)

A) **show ip interface**
B) **show ip pim interface**
C) **show ip pim neighbor**
D) **mrinfo**

Q10) Which command shows the multicast routing table? (Source: Multicast Configuration and Verification)

A) **show ip route**
B) **show ip pim route**
C) **show ip multicast route**
D) **show ip mroute**

Module Self-Check Answer Key

Q1)	A
Q2)	B
Q3)	A, D
Q4)	C
Q5)	C, D
Q6)	A, B
Q7)	B
Q8)	A
Q9)	C
Q10)	D

Module 8

Implementing IPv6

Overview

IP version 6 (IPv6) is a technology developed to overcome the limitations of the current standard, IP version 4 (IPv4), which allows end systems to communicate and forms the foundation of the Internet as we know it today.

One of the major shortcomings of IPv4 is its limited amount of address space. The explosion of new IP-enabled devices and the growth of undeveloped regions have fueled the need for more addresses.

In the United States, the Department of Defense (DoD) is a primary driver for the adoption of IPv6 and has set a date of 2008 for all systems to be migrated to the new standard. Currently, the other main market opportunities are the National Research and Education Network (NREN), government agencies, enterprises, service providers, home networking, consumer appliances, distributed online gaming, and wireless services.

Objectives

Upon completing this lesson, you will be able to describe how IPv6 functions and satisfies the increasingly complex requirements of hierarchical addressing. This ability includes being able to meet these objectives:

- Describe how IPv6 functions in order to satisfy the requirements of IPv6 addressing
- Describe IPv6 addressing
- Describe IPv6 addressing, neighbor discovery, and differences between IPv4 and IPv6
- Identify how you use IPv6 with OSPF
- Identify IPv6 integration and coexistence methods

Lesson 1

Introducing IPv6

Overview

The ability to scale networks for future demands requires a limitless supply of IP addresses and improved mobility. IP version 6 (IPv6) combines expanded addressing with a more efficient and feature-rich header to meet the demands for scalable networks in the future.

IPv6 satisfies the increasingly complex requirements of hierarchical addressing that IP version 4 (IPv4) does not provide. One key benefit is that IPv6 can recreate end-to-end communications without the need for Network Address Translation (NAT)—a requirement for a new generation of shared-experience and real-time applications.

Transitions to IPv6 from IPv4 deployments can use a variety of techniques, including an autoconfiguration function. This lesson describes the functionality and benefits of IPv6. Cisco Systems currently supports IPv6 in Cisco IOS Software Release 12.2(2)T and later.

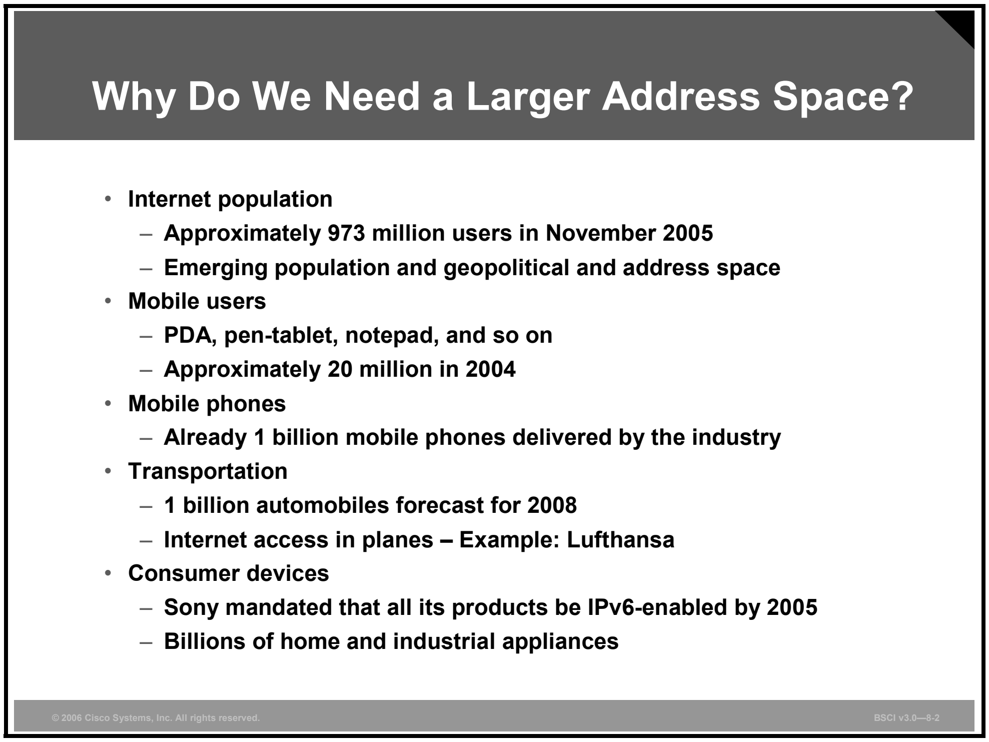
Objectives

Upon completing this lesson, you will be able to describe how IPv6 functions to satisfy the requirements of IPv6 addressing. This ability includes being able to meet these objectives:

- Explain how IPv6 deals with the limitations of IPv4
- Describe the features of IPv6 addressing

Explaining IPv6

This topic explains IPv6.



Why Do We Need a Larger Address Space?

- Internet population
  - Approximately 973 million users in November 2005
  - Emerging population and geopolitical and address space
- Mobile users
  - PDA, pen-tablet, notepad, and so on
  - Approximately 20 million in 2004
- Mobile phones
  - Already 1 billion mobile phones delivered by the industry
- Transportation
  - 1 billion automobiles forecast for 2008
  - Internet access in planes – Example: Lufthansa
- Consumer devices
  - Sony mandated that all its products be IPv6-enabled by 2005
  - Billions of home and industrial appliances

BSCI v3.0—8-2

The Internet will be transformed after IPv6 fully replaces its less versatile parent years from now. Nevertheless, IPv4 is in no danger of disappearing overnight. Rather, it will coexist with and then gradually be replaced by IPv6. This change has already begun, particularly in Europe, Japan, and Asia Pacific.

These areas are exhausting their allotted IPv4 addresses, which makes IPv6 all the more attractive. In addition to its technical and business potential, IPv6 offers a virtually unlimited supply of IP addresses. The existing IPv4 provides some 2 billion useable addresses with its 32-bit address space,

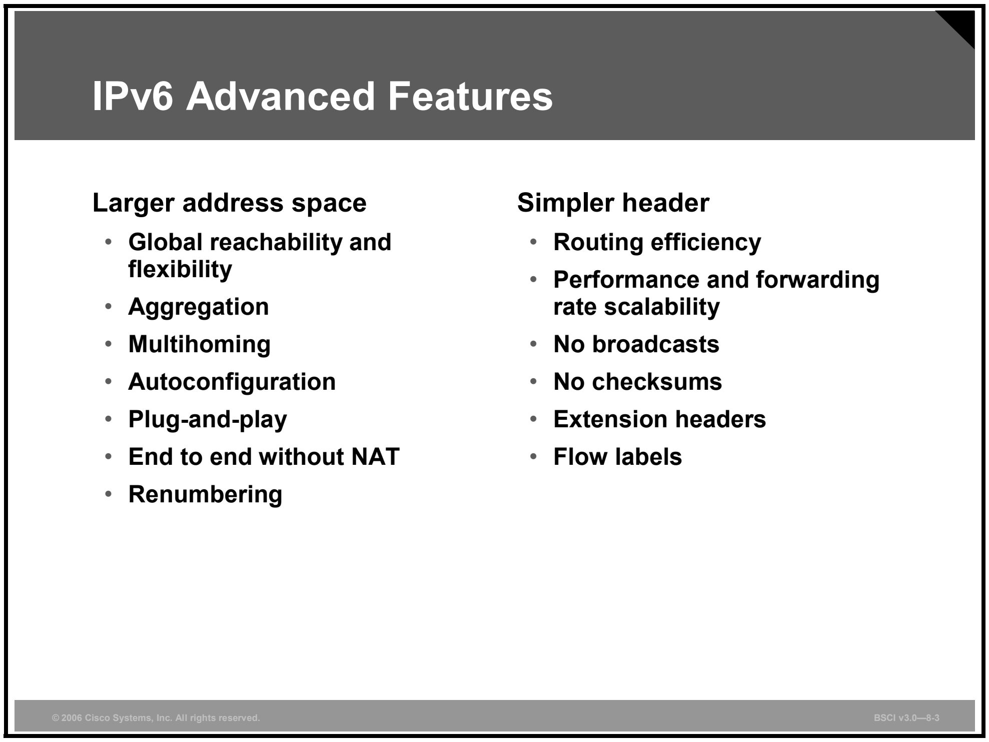
IPv6, because of its generous 128-bit address space, will generate a virtually unlimited stock of addresses—enough to allocate more than the entire IPv4 Internet address space to everyone on the planet.

Consequently, some countries, such as Japan, are aggressively adopting IPv6. Others, such as those in the European Union, are moving toward IPv6, and China is considering building pure IPv6 networks from the ground up.

As of October 1, 2003, even in North America, where Internet addresses are abundant, the U.S. Department of Defense (DoD) mandated that all new equipment purchased be IPv6-capable. In fact, the department intends to switch entirely to IPv6 equipment by 2008. As these examples illustrate, IPv6 enjoys strong momentum.

Describing IPv6 Features

This topic describes the features of IPv6.



IPv6 Advanced Features

Larger address space

- Global reachability and flexibility
- Aggregation
- Multihoming
- Autoconfiguration
- Plug-and-play
- End to end without NAT
- Renumbering

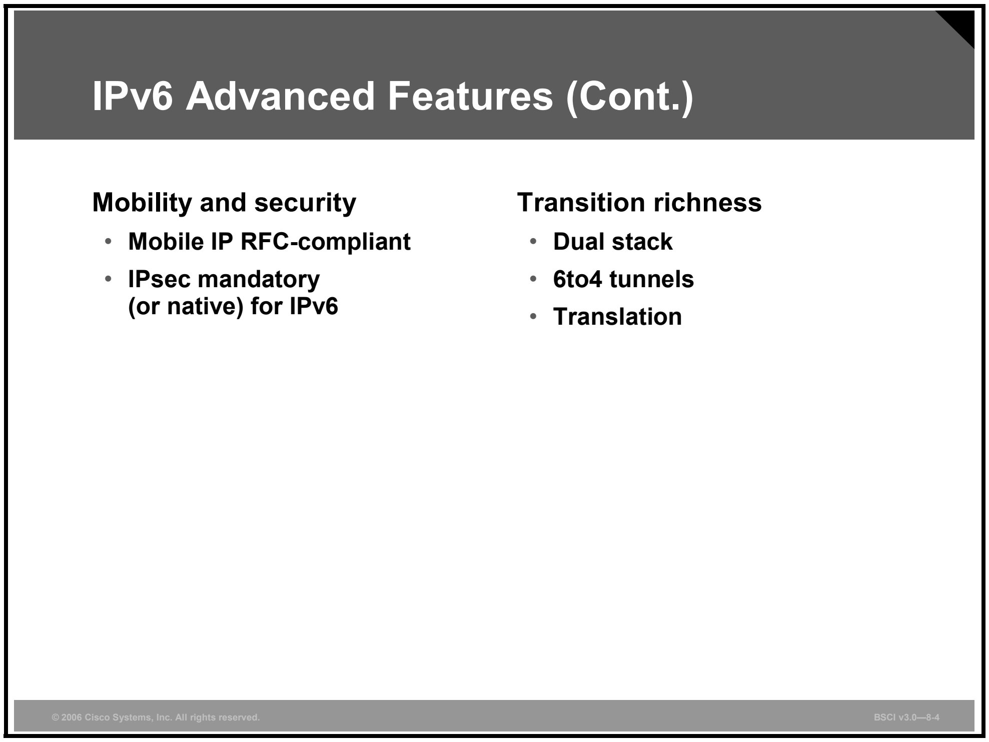
Simpler header

- Routing efficiency
- Performance and forwarding rate scalability
- No broadcasts
- No checksums
- Extension headers
- Flow labels

BSCI v3.0—8-3

IPv6 is a powerful enhancement to IPv4. There are several features in IPv6 that offer functional improvements. What IP developers learned from using IPv4 suggested changes to better suit current and foreseeable network demands:

- **Larger address space:** Larger address space includes several enhancements: improved global reachability and flexibility; the aggregation of prefixes that are announced in routing tables; multihoming to several Internet service providers (ISPs); autoconfiguration that can include link-layer addresses in the address space; plug-and-play options; and public-to private readdressing end to end without address translation; and simplified mechanisms for address renumbering and modification.
- **Simpler header:** A simpler header offers several advantages over IPv4: better routing efficiency for performance and forwarding-rate scalability; no broadcasts and thus no potential threat of broadcast storms; no requirement for processing checksums; simpler and more efficient extension header mechanisms; and flow labels for per-flow processing with no need to open the transport inner packet to identify the various traffic flows.



IPv6 Advanced Features (Cont.)

Mobility and security

- **Mobile IP RFC-compliant**
- **IPsec mandatory (or native) for IPv6**

Transition richness

- **Dual stack**
- **6to4 tunnels**
- **Translation**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-4

- **Mobility and security:** Mobility and security help ensure compliance with mobile IP and IPsec standards functionality. Mobility enables people to move around in networks with mobile network devices—with many having wireless connectivity.

  Mobile IP is an Internet Engineering Task Force (IETF) standard available for both IPv4 and IPv6. The standard enables mobile devices to move without breaks in established network connections. Because IPv4 does not automatically provide this kind of mobility, you must add it with additional configurations.

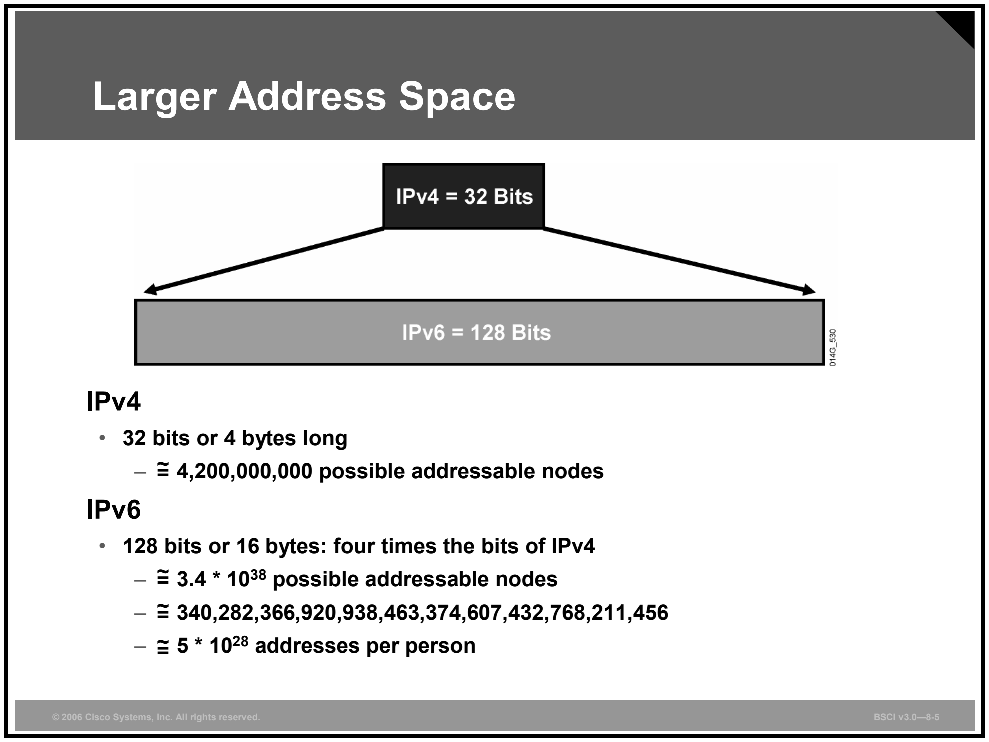
  In IPv6, mobility is built in, which means that any IPv6 node can use it when necessary. The routing headers of IPv6 make mobile IPv6 much more efficient for end nodes than mobile IPv4.

  IPsec is the IETF standard for IP network security, available for both IPv4 and IPv6. Although the functionalities are essentially identical in both environments, IPsec is mandatory in IPv6. IPsec is enabled on every IPv6 node and is available for use. The availability of IPsec on all nodes makes the IPv6 Internet more secure. IPsec also requires keys for each party, which implies a global key deployment and distribution.

- **Transition richness:** There are two ways to incorporate existing IPv4 capabilities with the added features of IPv6:

  — One approach is to have a dual stack with both IPv4 and IPv6 configured on the interface of a network device.

  — Another technique—called “IPv6 over IPv4” or “6to4” tunneling—uses an IPv4 tunnel to carry IPv6 traffic. This newer method (RFC 3056) replaces an older technique of IPv4-compatible tunneling (RFC 2893). Cisco IOS Software Release 12.3(2)T (and later) also allows protocol translation (NAT-PT) between IPv6 and IPv4. This translation allows direct communication between hosts speaking different protocols.



Larger Address Space

IPv4 = 32 Bits
IPv6 = 128 Bits
014G_530

IPv4

- **32 bits or 4 bytes long**
  - **≅ 4,200,000,000 possible addressable nodes**

IPv6

- **128 bits or 16 bytes: four times the bits of IPv4**
  - **≅ 3.4 * 10^{38} possible addressable nodes**
  - **≅ 340,282,366,920,938,463,374,607,432,768,211,456**
  - **≅ 5 * 10^{28} addresses per person**

BSCI v3.0—8-5

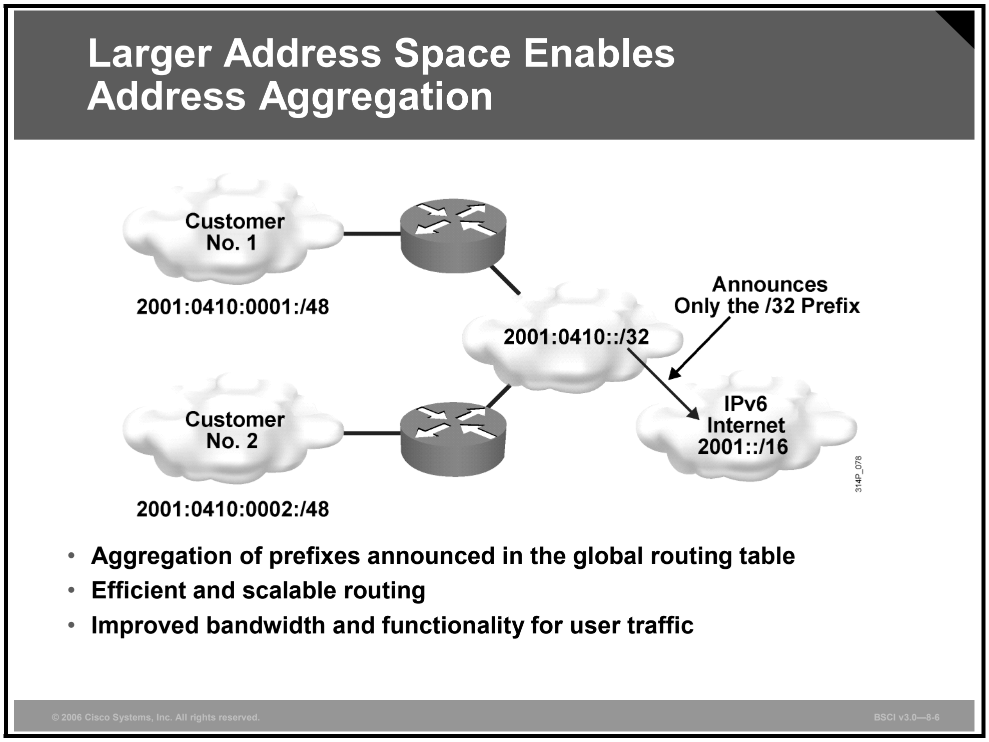
IPv6 increases the number of address bits by a factor of 4, from 32 to 128. This factor enables a very large number of addressable nodes; however, as in any addressing scheme, not all the addresses are used or available.

Current IPv4 protocol address use is extended by applying techniques such as NAT and temporary address allocations. But the manipulation of data payload by intermediate devices challenges (or complicates) the advantages of peer-to-peer communication, end-to-end security, and quality of service (QoS).

IPv6 gives every user multiple global addresses that can be used for a wide variety of devices, including cell phones, personal digital assistants (PDAs), and IP-enabled vehicles. Quadrupling the available 32-bit IPv4 address space to 128 bits, IPv6 addresses the need for always-on environments. These addresses are reachable without using IP address translation, pooling, and temporary allocation techniques.

Increasing the number of bits for the address also increases the IPv6 header size. Because each IP header contains a source and a destination address, the size of the header fields that contains the addresses is 256 bits for IPv6 compared to 64 bits for IPv4.

Note For more IETF information on IPv6 addressing details, refer to RFC 3513.



Larger Address Space Enables Address Aggregation

- Aggregation of prefixes announced in the global routing table
- Efficient and scalable routing
- Improved bandwidth and functionality for user traffic

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-6

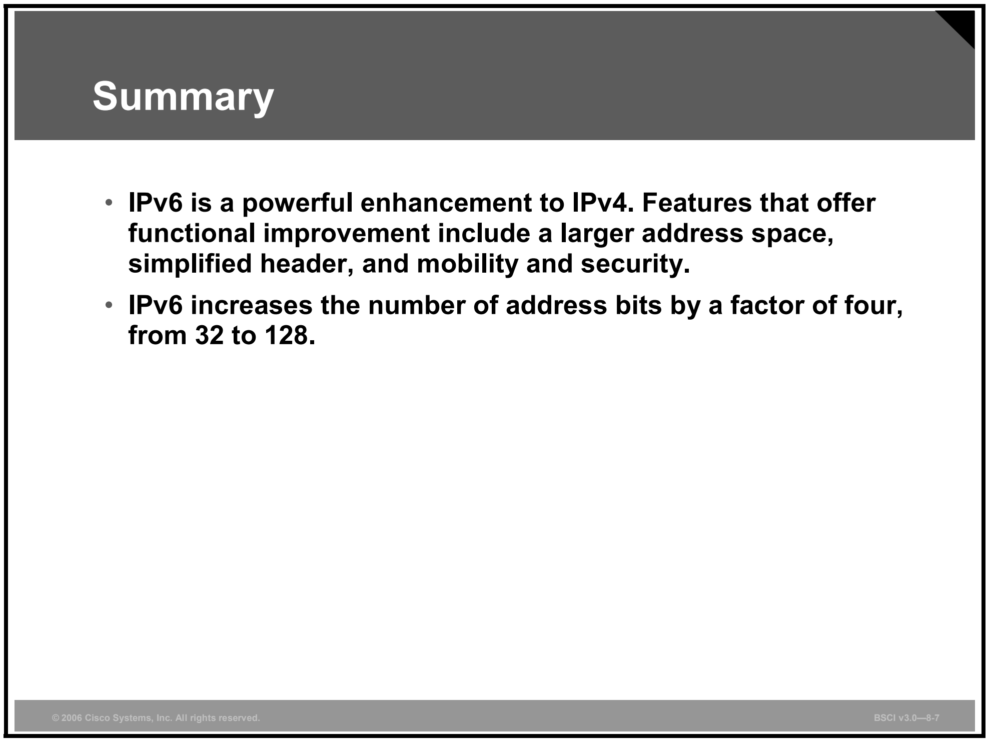
Larger address spaces make room for large address allocations to ISPs and organizations. An ISP aggregates all the prefixes of its customers into a single prefix and announces the single prefix to the IPv6 Internet. The increased address space is sufficient to allow organizations to define a single prefix for the entire network as well.

Aggregation of customer prefixes results in an efficient and scalable routing table. Scalable routing is necessary to expand broader adoption of network functions. Improved network bandwidth and functionality for user traffic will connect the various devices and applications. Internet usage—both now and in the future—can include these elements:

- A huge increase in the number of broadband consumers with high-speed, always-on connections
- Users who spend more time online and are generally willing to spend more money on communications services (such as music) and high-value searchable offerings
- Home networks with expanded network applications such as wireless Voice over IP (VoIP), home surveillance, and advanced services such as real-time video on demand (VoD)
- Massively scalable games with global participants and media-rich e-learning, providing learners with on-demand remote labs or lab simulations

Summary

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



Summary

- **IPv6 is a powerful enhancement to IPv4. Features that offer functional improvement include a larger address space, simplified header, and mobility and security.**
- **IPv6 increases the number of address bits by a factor of four, from 32 to 128.**

Lesson 2

Defining IPv6 Addressing

Overview

Before you can implement IP version 6 (IPv6) on a network, you must have a solid understanding of IPv6 addressing, packet structure, extension headers, and base features.

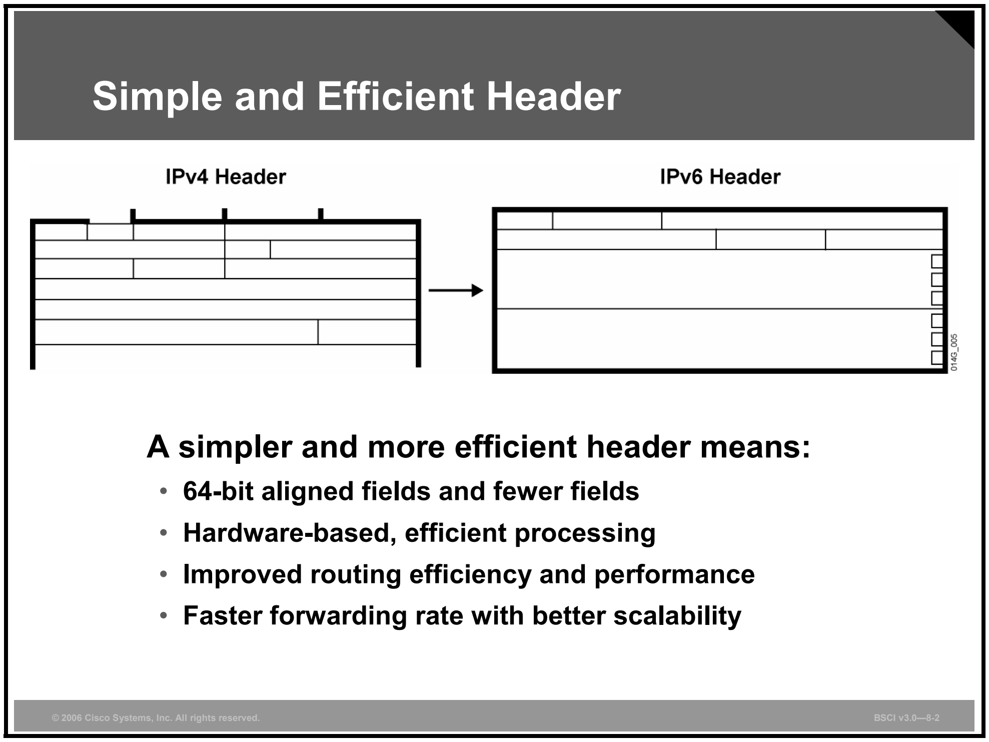
Objectives

Upon completing this lesson, you will be able to describe IPv6 addressing. This ability includes being able to meet these objectives:

- Describe the structure of IPv6 headers in terms of format and extension headers
- Show how an IPv6 address is represented
- Describe the three address types used in IPv6

Describing IPv6 Addressing Architecture

This topic provides an introduction to the IPv6 frame format.



Simple and Efficient Header

A simpler and more efficient header means:

- **64-bit aligned fields and fewer fields**
- **Hardware-based, efficient processing**
- **Improved routing efficiency and performance**
- **Faster forwarding rate with better scalability**

BSCI v3.0—8-2

The IP version 4 (IPv4) header contains 12 basic header fields, followed by an options field and a data portion (usually the transport layer segment). The basic IPv4 header has a fixed size of 20 octets. The variable-length options field increases the size of the total IP header. IPv6 contains 5 of the 12 IPv4 basic header fields. The IPv6 header does not require the other seven fields.

Routers handle fragmentation in IPv4, which causes a variety of processing issues. IPv6 routers no longer perform fragmentation. Instead, a discovery process is used to determine the optimum maximum transmission unit (MTU) to use during a given session.

In the discovery process, the source IPv6 device attempts to send a packet at the size that is specified by the upper IP layers, for example, the transport and application layers.

If the device receives an "ICMP packet too big" message, it retransmits the MTU discover packet with a smaller MTU and repeats the process until it gets a response that the discover packet arrived intact. Then it sets the MTU for the session.

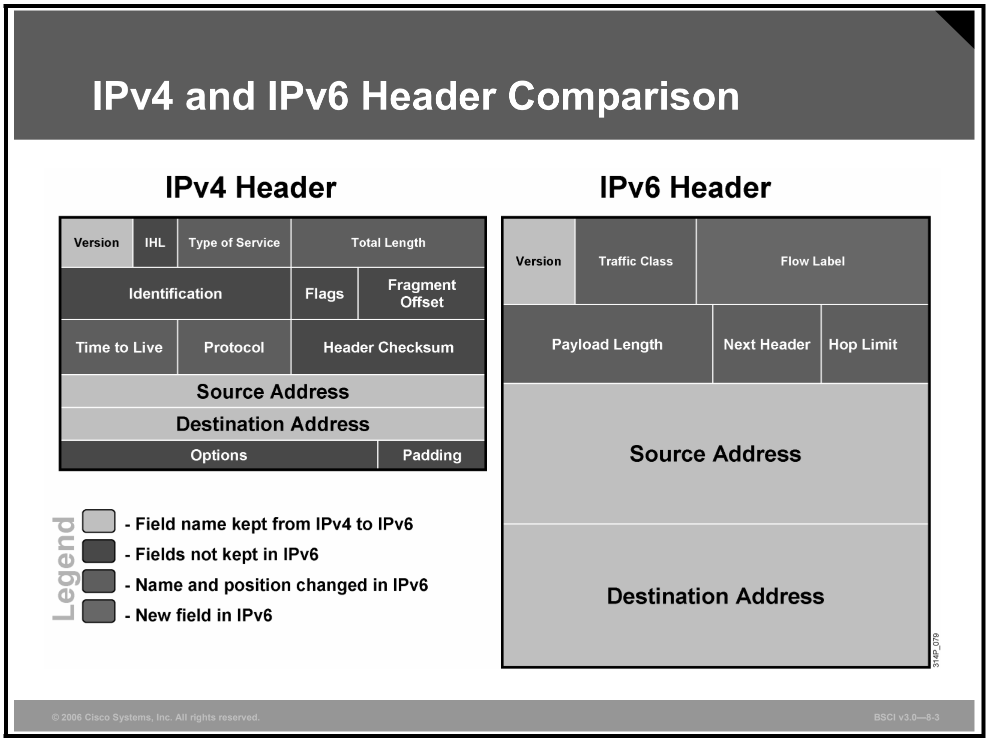
The "ICMP packet too big" message contains the proper MTU size for the pathway. Each source device needs to track the MTU size for each session. Generally, the tracking is done by creating a cache that is based on the destination address; however, it can also be done by using the flow label. If source-based routing is performed, the tracking of the MTU size can be done by using the source address.

The discovery process is beneficial because, as routing pathways change, a new MTU might be more appropriate. When a device receives an “ICMP packet too big” message, it decreases its MTU size if the Internet Control Message Protocol (ICMP) message contains a recommended MTU that is less than the current MTU of the device.

A device performs an MTU discovery every 5 minutes to see whether the MTU has increased along the pathway. Application and transport layers for IPv6 accept MTU reduction notifications from the IPv6 layer.

If they do not accept the notifications, IPv6 has a mechanism to fragment packets that are too large; however, upper layers are encouraged to avoid sending messages that require fragmentation.

Link-layer technologies already perform checksum and error control. Because link-layer technologies are relatively reliable, an IP header checksum is considered to be redundant. Without the IP header checksum, the upper-layer optional checksums, such as User Datagram Protocol (UDP), are now mandatory.

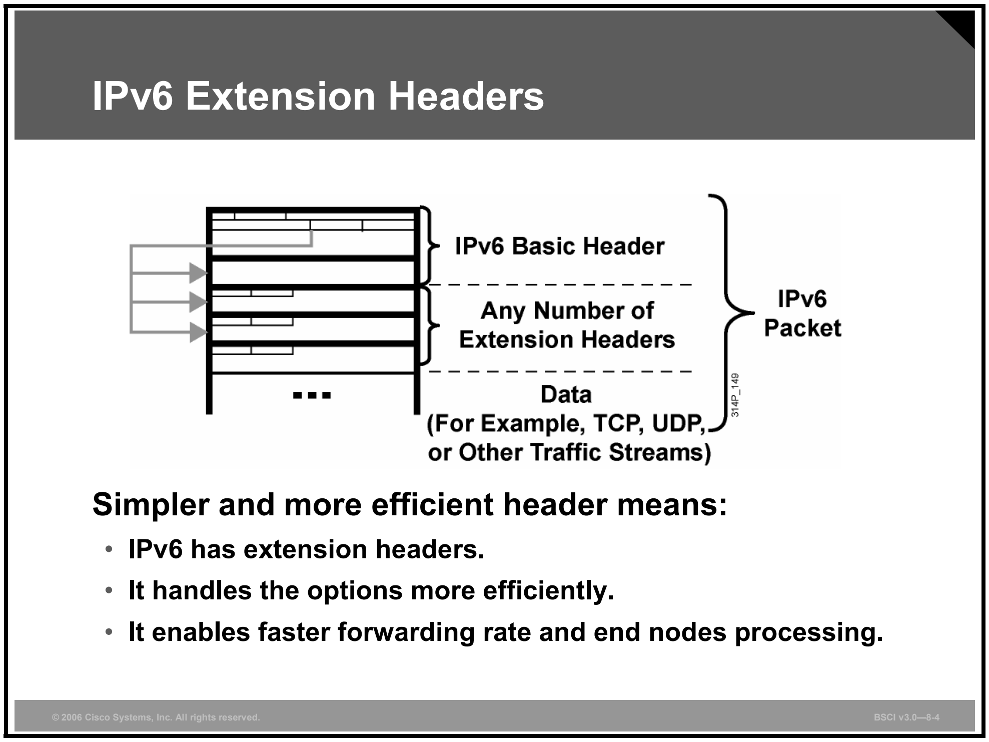


IPv4 and IPv6 Header Comparison

The IPv6 header has 40 octets in contrast to the 20 octets in IPv4. IPv6 has a smaller number of fields, and the header is 64-bit aligned to enable fast processing by current processors. Address fields are four times larger than in IPv4.

The IPv6 header contains these fields:

- **Version:** A 4-bit field, the same as in IPv4. It contains the number 6 instead of the number 4 for IPv4.
- **Traffic Class:** An 8-bit field similar to the type of service (ToS) field in IPv4. It tags the packet with a traffic class that it uses in differentiated services (DiffServ). These functionalities are the same for IPv6 and IPv4.
- **Flow Label:** A completely new 20-bit field. It tags a flow for the IP packets. It can be used for multilayer switching techniques and faster packet-switching performance.
- **Payload Length:** Similar to the Total Length field of IPv4.
- **Next Header:** The value of this field determines the type of information that follows the basic IPv6 header. It can be a transport-layer packet, such as TCP or UDP, or it can be an extension header. The next header field is similar to the Protocol field of IPv4.
- **Hop Limit:** This field specifies the maximum number of hops that an IP packet can traverse. Each hop or router decreases this field by one (similar to the Time to Live [TTL] field in IPv4). Because there is no checksum in the IPv6 header, the router can decrease the field without recomputing the checksum. On IPv4 routers the recomputation costs processing time.
- **Source Address:** This field has 16 octets or 128 bits. It identifies the source of the packet.
- **Destination Address:** This field has 16 octets or 128 bits. It identifies the destination of the packet.
- **Extension Headers:** The extension headers, if any, and the data portion of the packet follow the eight fields. The number of extension headers is not fixed, so the total length of the extension header chain is variable.



IPv6 Extension Headers

IPv6 Basic Header

Any Number of Extension Headers

Data (For Example, TCP, UDP, or Other Traffic Streams)

IPv6 Packet

Simpler and more efficient header means:

- **IPv6 has extension headers.**
- **It handles the options more efficiently.**
- **It enables faster forwarding rate and end nodes processing.**

BSCI v3.0—8-4

There are many types of extension headers. When multiple extension headers are used in the same packet, the order of the headers should be as follows:

1. **IPv6 header:** This header is the basic header described in the previous figure.

2. **Hop-by-hop options header:** When this header is used for the router alert (Resource Reservation Protocol [RSVP] and Multicast Listener Discovery version 1 [MLDv1]) and the jumbogram, this header (value = 0) is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the basic IPv6 packet header.

3. **Destination options header (when the routing header is used):** This header (value = 60) can follow any hop-by-hop options header, in which case the destination options header is processed at the final destination and also at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header, in which case the destination options header is processed only at the final destination. For example, mobile IP uses this header.

4. **Routing header:** This header (value = 43) is used for source routing and mobile IPv6.

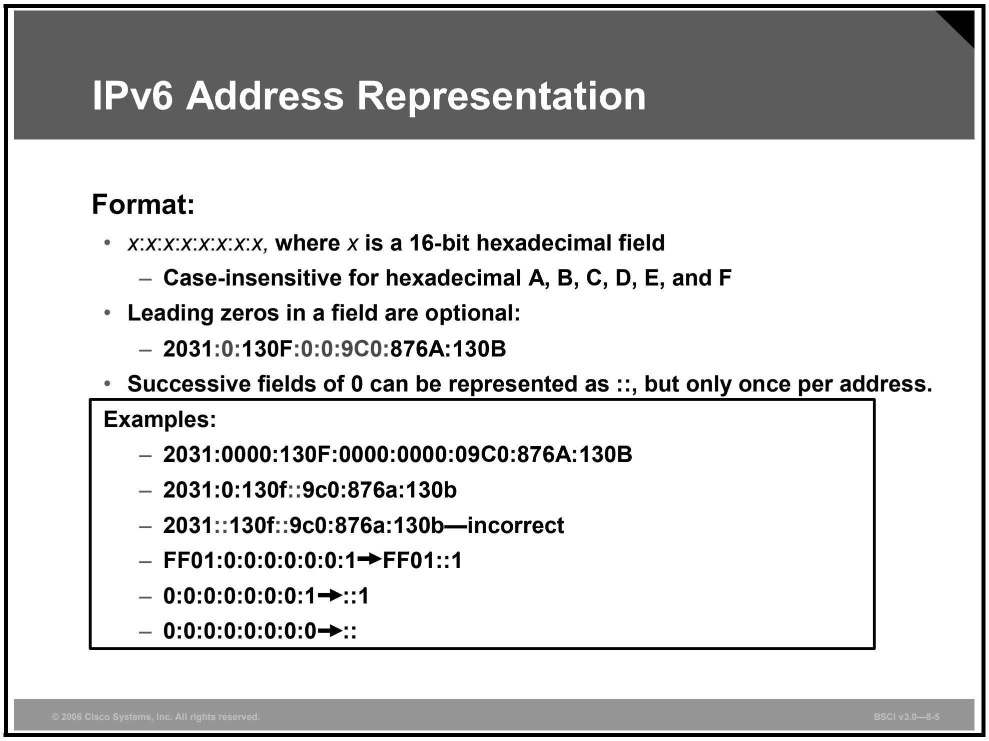
5. **Fragment header:** This header is used when a source must fragment a packet that is larger than the MTU for the path between itself and a destination device. The fragment header is used in each fragmented packet.

6. **Authentication header and Encapsulating Security Payload header:** The authentication header (value = 51) and the ESP header (value = 50) are used within IPsec to provide authentication, integrity, and confidentiality of a packet. These headers are identical for both IPv4 and IPv6.

7. **Upper-layer header:** The upper-layer (transport) headers are the typical headers used inside a packet to transport the data. The two main transport protocols are TCP (value = 6) and UDP (value = 17).

Defining Address Representation

This topic describes how IPv6 addresses are shown in hexadecimal and the appropriate methods for abbreviation of addresses.



IPv6 Address Representation

Format:

- *x*:*x*:*x*:*x*:*x*:*x*:*x*:*x*, **where *x* is a 16-bit hexadecimal field**
  - **Case-insensitive for hexadecimal A, B, C, D, E, and F**
- **Leading zeros in a field are optional:**
  - **2031:0:130F:0:0:9C0:876A:130B**
- **Successive fields of 0 can be represented as ::, but only once per address.**

Examples:

- **2031:0000:130F:0000:0000:09C0:876A:130B**
- **2031:0:130f::9c0:876a:130b**
- **2031::130f::9c0:876a:130b—incorrect**
- **FF01:0:0:0:0:0:0:1→FF01::1**
- **0:0:0:0:0:0:0:1→::1**
- **0:0:0:0:0:0:0:0→::**

BSCI v3.0—8-5

Colons separate entries in a series of 16-bit hexadecimal fields that represent IPv6 addresses. The hexadecimal digits A, B, C, D, E, and F represented in IPv6 are case-insensitive.

IPv6 does not require explicit address string notation. Use the following guidelines for IPv6 address string notations:

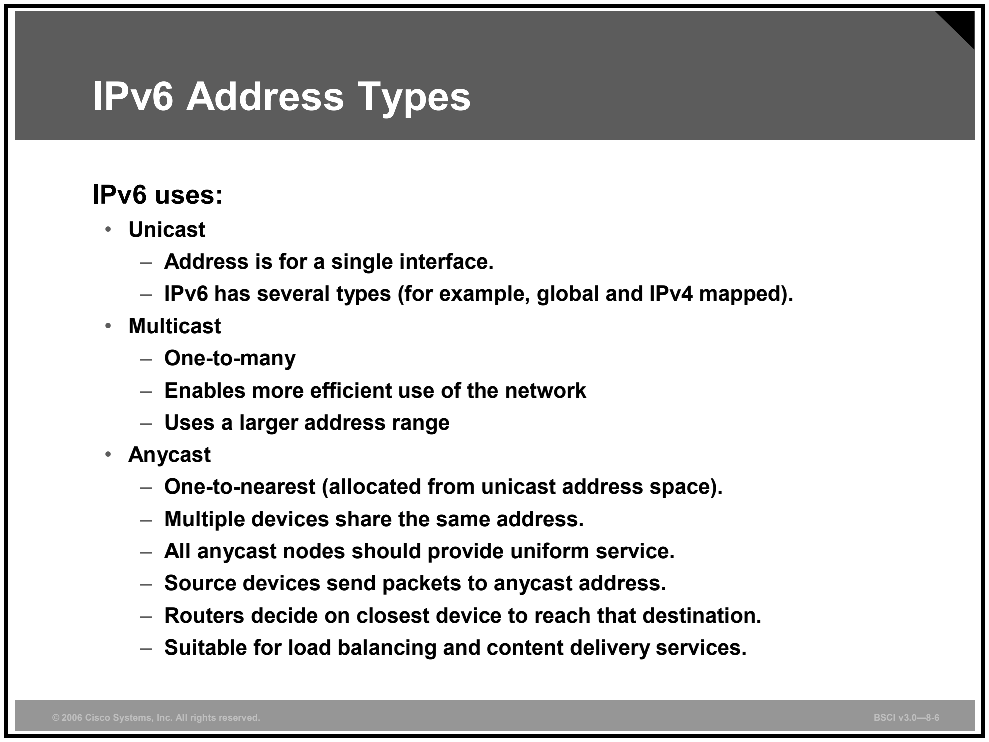
- The leading zeros in a field are optional, so that 09C0 = 9C0 and 0000 = 0.
- Successive fields of zeros can be represented as “::” only once in an address.
- An unspecified address is written as “::” because it contains only zeros.

Using the “::” notation greatly reduces the size of most addresses. For example, FF01:0:0:0:0:0:0:1 becomes FF01::1.

Note An address parser identifies the number of missing zeros by separating the two parts and entering 0 until the 128 bits are complete. If two “::” notations are placed in the address, there is no way to identify the size of each block of zeros.

IPv6 Address Types

This topic describes the three address types used in IPv6.



IPv6 Address Types

IPv6 uses:

- **Unicast**
  - **Address is for a single interface.**
  - **IPv6 has several types (for example, global and IPv4 mapped).**
- **Multicast**
  - **One-to-many**
  - **Enables more efficient use of the network**
  - **Uses a larger address range**
- **Anycast**
  - **One-to-nearest (allocated from unicast address space).**
  - **Multiple devices share the same address.**
  - **All anycast nodes should provide uniform service.**
  - **Source devices send packets to anycast address.**
  - **Routers decide on closest device to reach that destination.**
  - **Suitable for load balancing and content delivery services.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-6

Broadcasting in IPv4 results in a number of problems. Broadcasting generates a number of interrupts in every computer on the network and, in some cases, triggers malfunctions that can completely halt an entire network. This disastrous network event is known as a "broadcast storm."

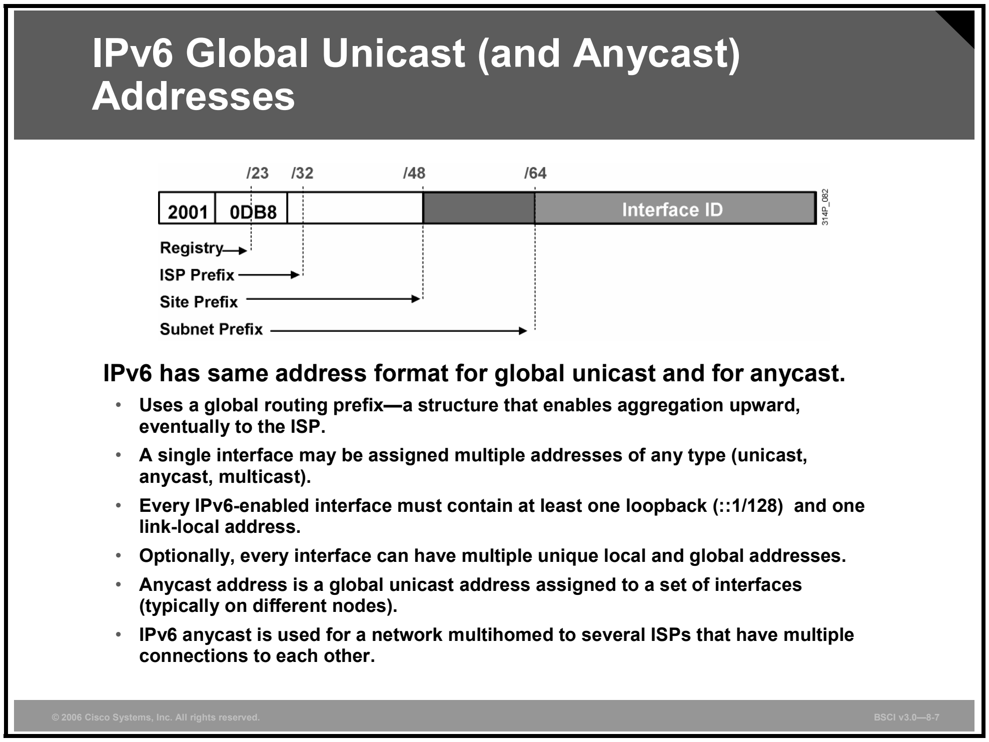
In IPv6, broadcasting does not exist. Broadcasts are replaced by multicasts and anycasts. Multicast enables efficient network operation by using a number of functionally specific multicast groups to send requests to a limited number of computers on the network. The multicast groups prevent most of the problems that are related to broadcast storms in IPv4.

The range of multicast addresses in IPv6 is larger than in IPv4. For the foreseeable future, allocation of multicast groups is not being limited.

IPv6 also defines a new type of address called an anycast address. An anycast address identifies a list of devices or nodes; therefore, an anycast address identifies multiple interfaces. A packet sent to an anycast address is delivered to the closest interface—as defined by the routing protocols in use—identified by the anycast address.

Anycast addresses are syntactically indistinguishable from global unicast addresses because anycast addresses are allocated from the global unicast address space.

Note Anycast addresses must not be used as the source address of an IPv6 packet.



/23
/32
/48
/64
2001
0DB8
Interface ID
314P_082
Registry
ISP Prefix
Site Prefix
Subnet Prefix

IPv6 Global Unicast (and Anycast) Addresses

IPv6 has same address format for global unicast and for anycast.

- Uses a global routing prefix—a structure that enables aggregation upward, eventually to the ISP.
- A single interface may be assigned multiple addresses of any type (unicast, anycast, multicast).
- Every IPv6-enabled interface must contain at least one loopback (::1/128) and one link-local address.
- Optionally, every interface can have multiple unique local and global addresses.
- Anycast address is a global unicast address assigned to a set of interfaces (typically on different nodes).
- IPv6 anycast is used for a network multihomed to several ISPs that have multiple connections to each other.

BSCI v3.0—8-7

The global unicast and the anycast share the same address format. The unicast address space allocates the anycast addresses. To devices that are not configured for anycast, these addresses appear as unicast addresses.

When a unicast address is assigned to more than one interface, thus turning it into an anycast address, the nodes to which the address is assigned must be explicitly configured to use and recognize the anycast address.

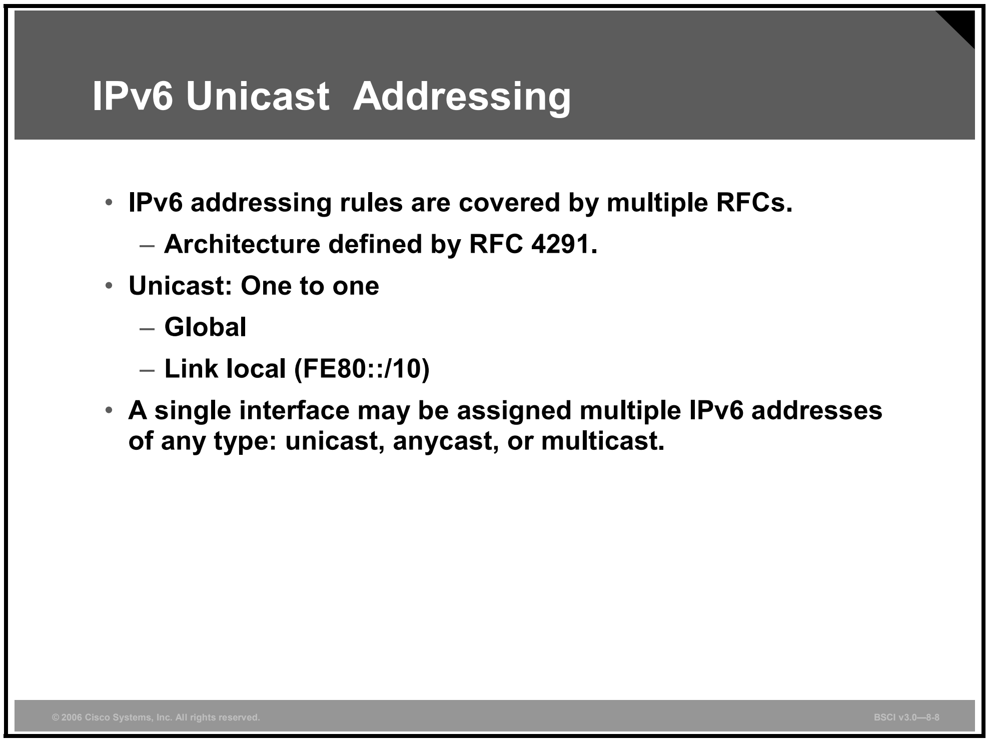
A packet that is sent to an anycast address routes to the closest device or interface that shares this address. A sender creates a packet with the anycast as the destination address and forwards it to its nearest router. The source can use the anycast addresses to control the pathway across which traffic flows.

Examples: Multiple ISPs and LANs with Multiple Routers

An example of anycast use in a Border Gateway Protocol (BGP) multihomed network is when a customer has multiple ISPs with multiple connections to one another. The customer can configure a different anycast address for each ISP.

Each router for the given ISP has the same configured anycast address. The source device can choose which ISP to send the packet to; however, the routers along the path determine the closest router to reach that ISP using the IPv6 anycast address.

Another use for an anycast is when a LAN is attached to multiple routers. These routers can have the same IPv6 anycast address so that distant devices need to identify only the anycast address. Intermediate devices can choose the best pathway to reach the closest entry point to that subnet.



IPv6 Unicast Addressing

- **IPv6 addressing rules are covered by multiple RFCs.**
  - **Architecture defined by RFC 4291.**
- **Unicast: One to one**
  - **Global**
  - **Link local (FE80::/10)**
- **A single interface may be assigned multiple IPv6 addresses of any type: unicast, anycast, or multicast.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-8

The IPv6 global unicast address is the equivalent of the IPv4 global unicast address. A global unicast address is an IPv6 address from the global unicast prefix. The structure of global unicast addresses enables aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Global unicast addresses used on links are aggregated upward through organizations and eventually to the Internet service providers (ISPs).

Global unicast addresses are defined by a global routing prefix, a subnet ID, and an interface ID. The IPv6 unicast address space encompasses the entire IPv6 address range, with the exception of FF00::/8 (1111 1111), which is used for multicast addresses. The current global unicast address assignment by the Internet Assigned Numbers Authority (IANA) uses the range of addresses that start with binary value 001 (2000::/3), which is one-eighth of the total IPv6 address space and is the largest block of assigned block addresses.

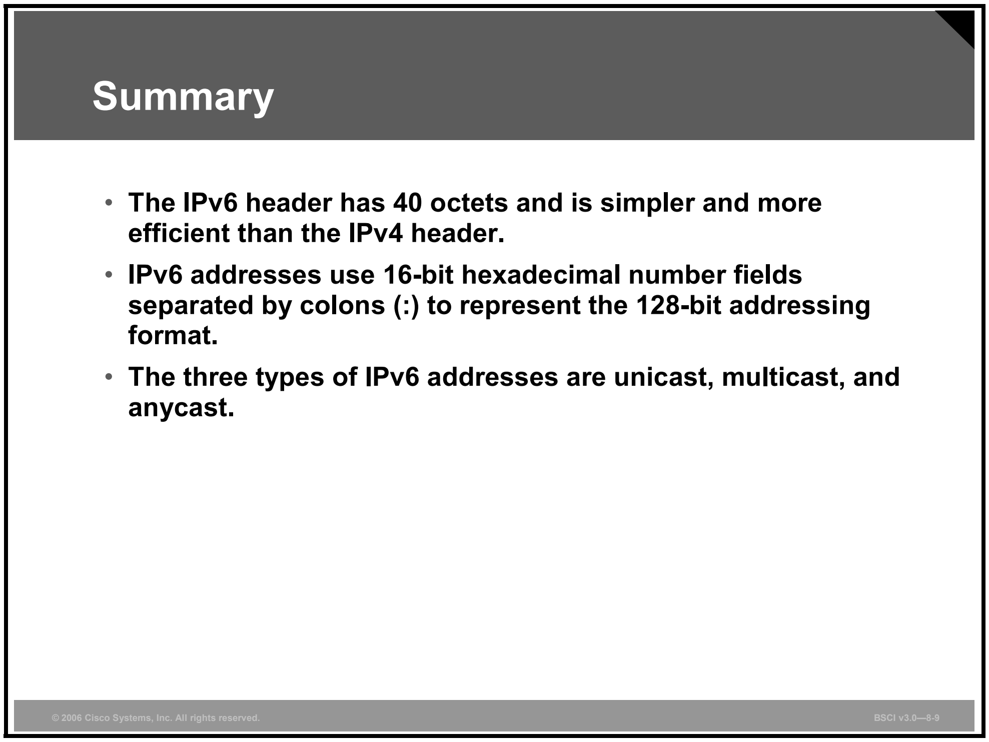
Addresses with a prefix of 2000::/3 (001) through E000::/3 (111), with the exception of the FF00::/8 (1111 1111) multicast addresses, are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format.

The IANA is allocating the IPv6 address space in the ranges of 2001::/16 to the registries.

The global unicast address typically consists of a 48-bit global routing prefix and a 16-bit subnet ID. In the now obsolete RFC 2374, *An IPv6 Aggregatable Global Unicast Address Format*, the global routing prefix included two other hierarchically structured fields called Top-Level Aggregator and Next-Level Aggregator. Because these fields were policy-based, the Internet Engineering Task Force (IETF) decided to remove them from the RFCs. However, some existing IPv6 networks deployed in the early days might still be using networks based on the older architecture. A 16-bit subnet field called Subnet ID could be used by individual organizations to create their own local addressing hierarchy and to identify subnets. This field allows an organization to use up to 65,535 individual subnets. (RFC 2374 has now been replaced by RFC 3587, *IPv6 Aggregatable Global Unicast Address Format.*)

Summary

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



Summary

- **The IPv6 header has 40 octets and is simpler and more efficient than the IPv4 header.**
- **IPv6 addresses use 16-bit hexadecimal number fields separated by colons (:) to represent the 128-bit addressing format.**
- **The three types of IPv6 addresses are unicast, multicast, and anycast.**

Lesson 3

Implementing Dynamic IPv6 Addresses

Overview

IP version 6 (IPv6) is a technology developed to overcome the limitations of the current standard, IP version 4 (IPv4). IPv6 is integrates the MAC address into the overall addressing scheme. IPv6 offers several autoconfiguration options and an improved method for implementing mobile IP.

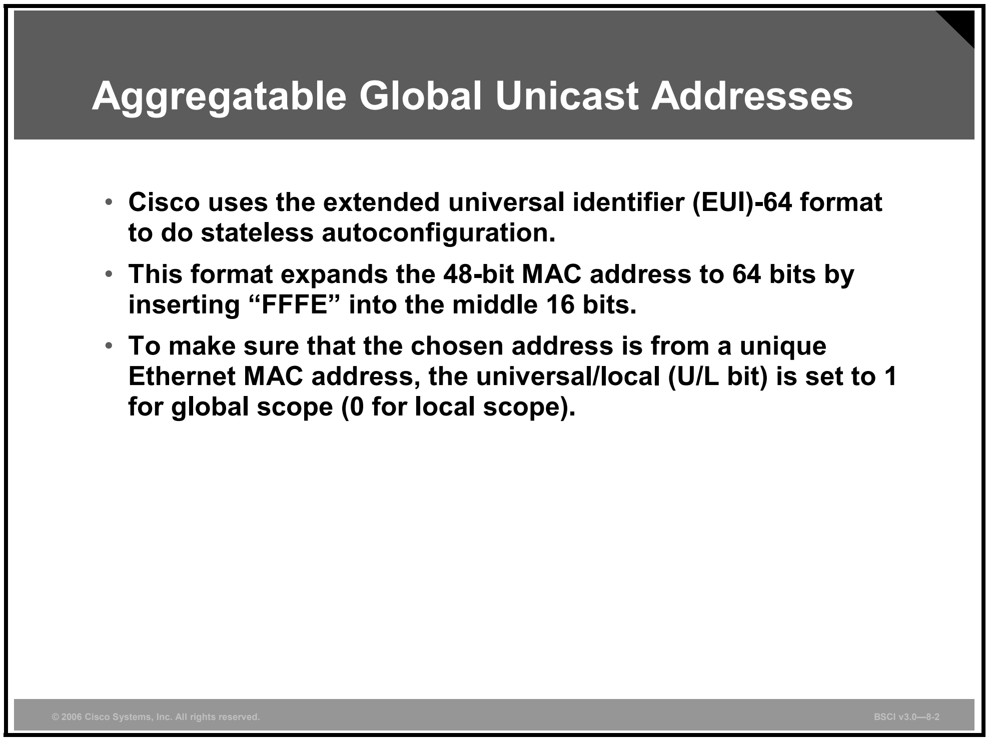
Objectives

Upon completing this lesson, you will be able to describe IPv6 addressing, neighbor discovery, and the differences between IPv4 and IPv6. This ability includes being able to meet these objectives:

- Explain how Ethernet MAC addresses can be used to generate a 64-bit interface ID for the host
- Explain how IPv6 improves multicast
- Describe how IPv6 simplifies mobile IP connections

Defining Host Interface Addresses

This topic explains how Ethernet MAC addresses can be used to generate a 64-bit interface ID for the host.



Aggregatable Global Unicast Addresses

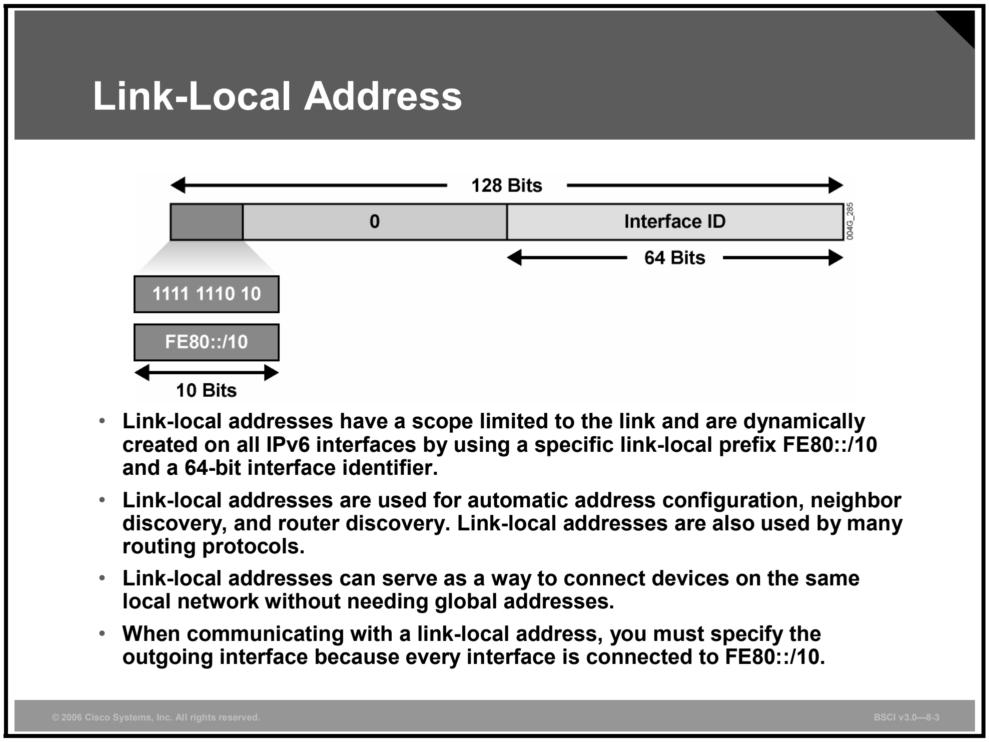
- **Cisco uses the extended universal identifier (EUI)-64 format to do stateless autoconfiguration.**
- **This format expands the 48-bit MAC address to 64 bits by inserting “FFFE” into the middle 16 bits.**
- **To make sure that the chosen address is from a unique Ethernet MAC address, the universal/local (U/L bit) is set to 1 for global scope (0 for local scope).**

BSCI v3.0—8-2

Use of EUI-64 Format in IPv6 Addresses

The 64-bit interface identifier in an IPv6 address is used to identify a unique interface on a link. A link is a network medium over which network nodes communicate using the link layer. The interface identifier may also be unique over a broader scope. In many cases, an interface identifier will be the same as or is based on the link-layer (MAC) address of an interface. As in IPv4, a subnet prefix in IPv6 is associated with one link.

Interface identifiers used in global unicast and other IPv6 address types must be 64 bits long and constructed in the Extended Universal Identifier (EUI)-64 format. The EUI-64 format interface ID is derived from the 48-bit link-layer (MAC) address by inserting the hexadecimal number FFFE between the upper 3 bytes (Organizational Unique Identifier [OUI] field) and the lower 3 bytes (serial number) of the link layer address. To make sure that the chosen address is from a unique Ethernet MAC address, the seventh bit in the high-order byte is set to 1 (equivalent to the IEEE G/L bit) to indicate the uniqueness of the 48-bit address.



Link-Local Address

- Link-local addresses have a scope limited to the link and are dynamically created 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. Link-local addresses are also used by many routing protocols.
- Link-local addresses can serve as a way to connect devices on the same local network without needing global addresses.
- When communicating with a link-local address, you must specify the outgoing interface because every interface is connected to FE80::/10.

BSCI v3.0—8-3

Interface identifiers in IPv6 addresses are used to identify interfaces on a link. They may also be thought of as the “host portion” of an IPv6 address.

Interface identifiers are required to be unique on that link.

Interface identifiers may also be unique over a broader scope: When the interface identifier is derived directly from the data link layer address of the interface (for example, IEEE 802.1Q MAC), the scope of that identifier is assumed to be universal (global).

Interface identifiers are always 64 bits and are dynamically created based on Layer 2 media and encapsulation.

IPv6 over Data Link Layers

IPv6 is defined on most of the current data link layers, including the following:

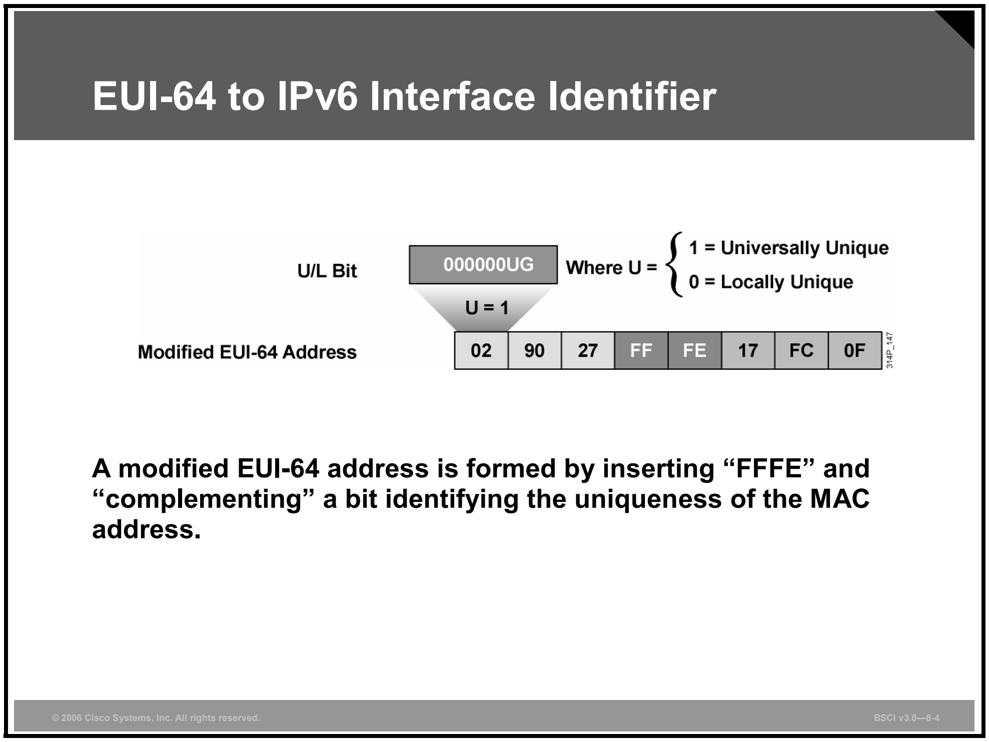
- Ethernet*
- PPP*
- High-Level Data Link Control (HDLC)*
- FDDI
- Token Ring
- Attached Resource Computer Network (ARCNET)
- Nonbroadcast multiaccess (NBMA)
- ATM**
- Frame Relay***
- IEEE 1394

* Cisco supports these data link layers.

** Cisco supports only ATM permanent virtual circuit (PVC) and ATM LAN Emulation (LANE).

*** Cisco supports only Frame Relay PVC.

An RFC describes the behavior of IPv6 in each of these specific data link layers, but Cisco IOS software does not necessarily support all of them. The data link layer defines how IPv6 interface identifiers are created and how neighbor discovery deals with data link layer address resolution.



EUI-64 to IPv6 Interface Identifier

U/L Bit

000000UG

Where U = 1 = Universally Unique; 0 = Locally Unique

U = 1

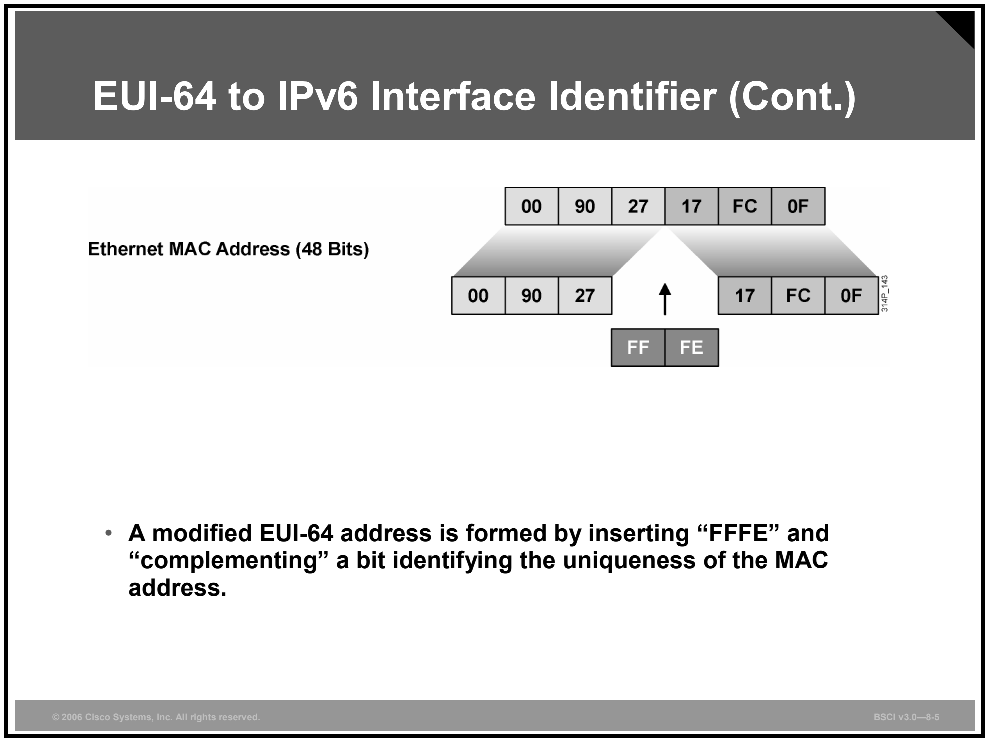
Modified EUI-64 Address

02	90	27	FF	FE	17	FC	0F

A modified EUI-64 address is formed by inserting “FFFE” and “complementing” a bit identifying the uniqueness of the MAC address.

EUI-64 to IPv6 Interface Identifier

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

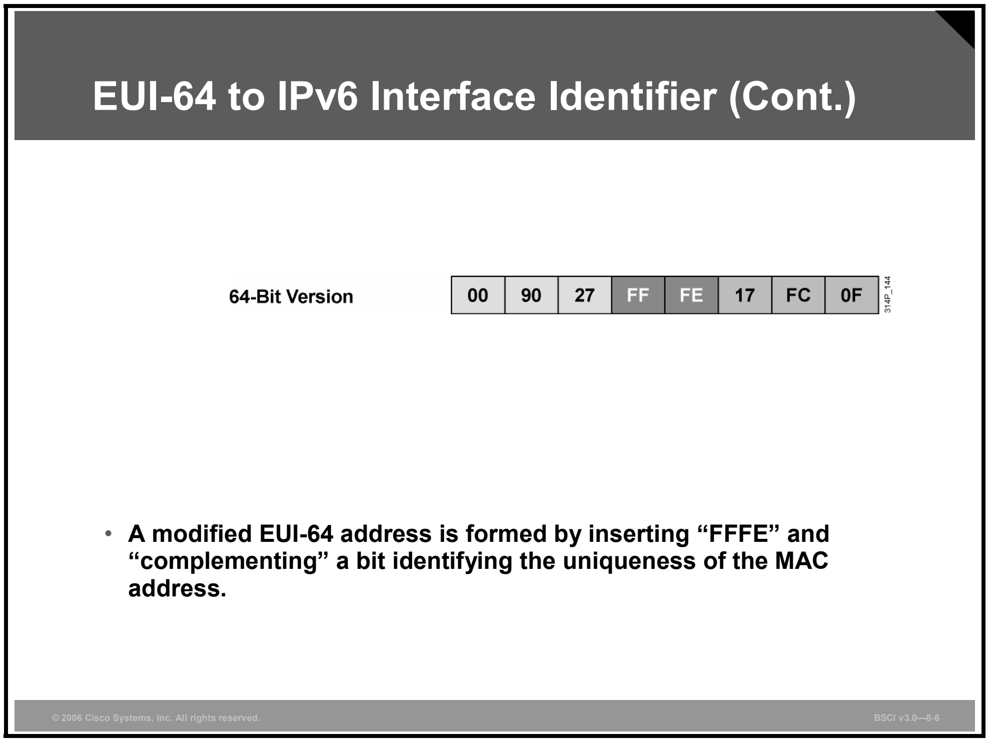


EUI-64 to IPv6 Interface Identifier (Cont.)

- **A modified EUI-64 address is formed by inserting “FFFE” and “complementing” a bit identifying the uniqueness of the MAC address.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-5

The seventh bit in an IPv6 interface identifier is referred to as the universal/local bit, or U/L bit. This bit identifies whether this interface identifier is locally unique on the link or whether it is universally unique. When the interface identifier is created from an Ethernet MAC address, it is assumed that the MAC address is universally unique and, therefore, that the interface identifier is universally unique.



EUI-64 to IPv6 Interface Identifier (Cont.)

- **A modified EUI-64 address is formed by inserting “FFFE” and “complementing” a bit identifying the uniqueness of the MAC address.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-6

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

The eighth bit, also known as the G bit, is the group/individual bit, for managing groups.

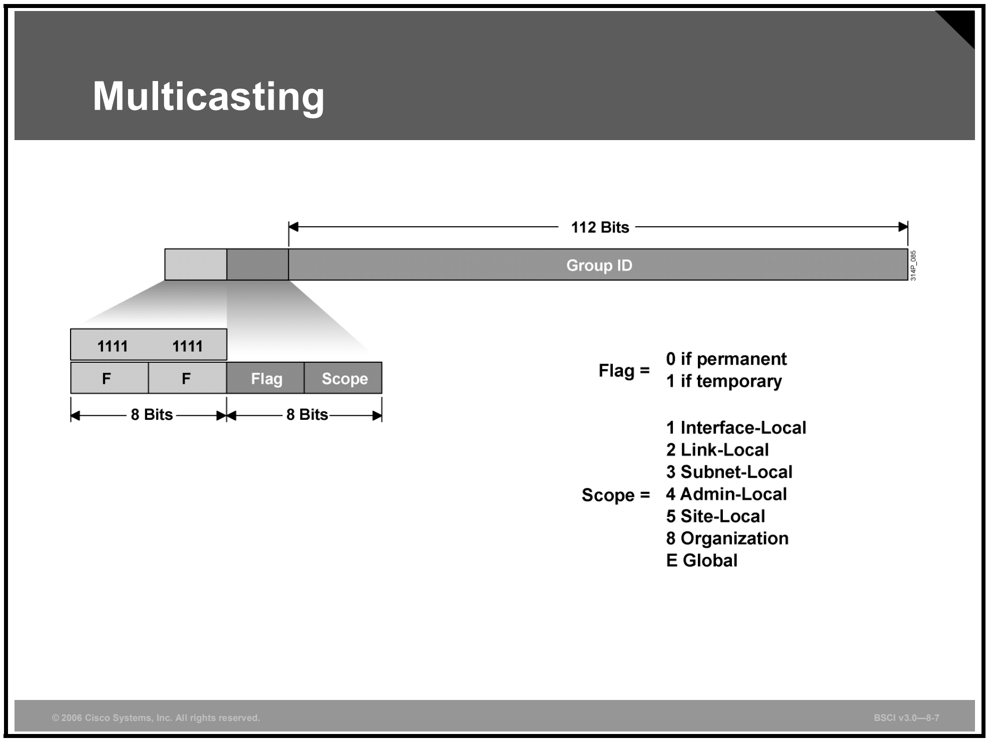
Because of certain privacy and security concerns, the implementation of autoconfiguration by a host may also create a random interface identifier using the MAC address as a base. This is considered a privacy extension because, without it, creating an interface identifier from a MAC address provides the ability to track the activity and point of connection.

Microsoft Windows XP is currently the only known implementation of this capability and prefers to use this address for outgoing communication, because the address has a short lifetime and will be regenerated periodically.

Note This process is defined in RFC 3041.

Explaining IPv6 Multicast

This topic explains how IPv6 improves multicast.



Multicasting

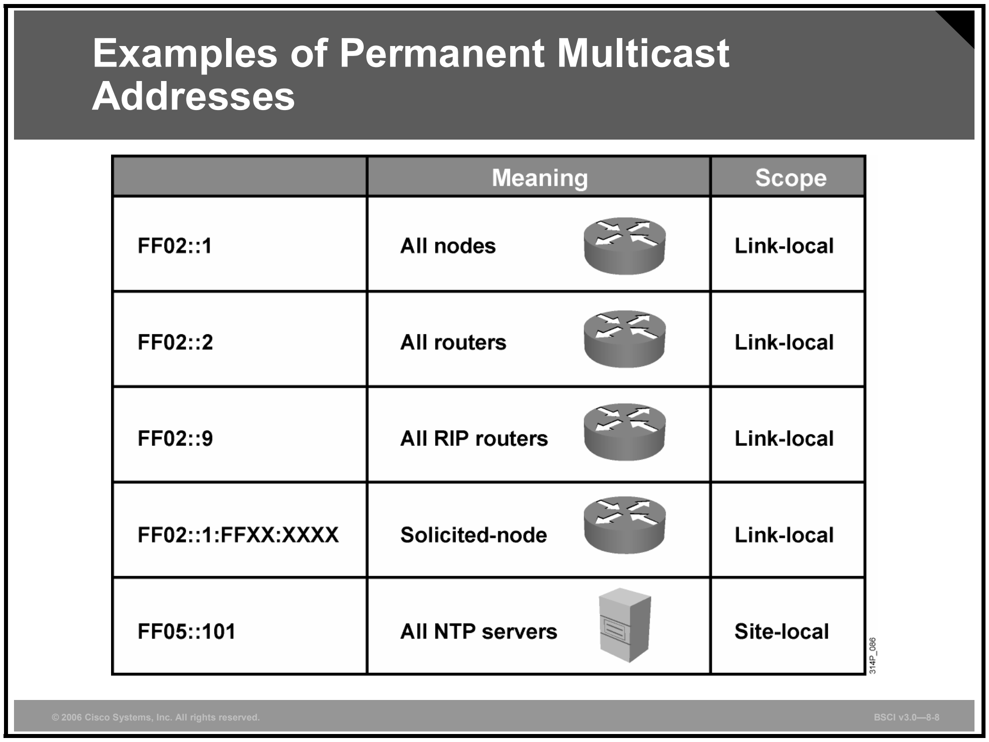
Flag = 0 if permanent
1 if temporary

Scope = 1 Interface-Local
2 Link-Local
3 Subnet-Local
4 Admin-Local
5 Site-Local
8 Organization
E Global

BSCI v3.0—8-7

A multicast address identifies a group of interfaces. Traffic sent to a multicast address travels to multiple destinations at the same time. An interface may belong to any number of multicast groups. Multicasting is extremely important to IPv6, because it is at the core of many IPv6 functions.

- IPv6 multicast addresses are defined by the prefix FF00::/8. The second octet defines the lifetime (flag) and the scope of the multicast address.
  - The flag parameter is equal to 0 for a permanent, or well-known, multicast address. For a temporary multicast address, the flag is equal to 1.
  - The scope parameter is equal to 1 for the scope of the interface (loopback transmission), 2 for the link scope (similar to unicast link-local scope), 3 for subnet-local scope where subnets may span multiple links, 4 for admin-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 starting with FF02::/16 is a permanent multicast address with a link-local scope.
- The multicast group ID consists of the lower 112 bits of the multicast address.
- 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.



Examples of Permanent Multicast Addresses

	Meaning	Scope
FF02::1	All nodes	Link-local
FF02::2	All routers	Link-local
FF02::9	All RIP routers	Link-local
FF02::1:FFXX:XXXX	Solicited-node	Link-local
FF05::101	All NTP servers	Site-local

314P_086

BSCI v3.0—8-8

The multicast addresses, FF00:: to FF0F::, are reserved. Within that range, the following are some examples of assigned addresses (there are many more assignments made; assignments are tracked by the Internet Assigned Numbers Authority [IANA]):

- **FF02::1** — All nodes on link (link-local scope)
- **FF02::2** — All routers on link
- **FF02::9** — All Routing Information Protocol (RIP) routers on link
- **FF02::1:FF*XX*:*XXXX*** — Solicited-node multicast on link, where *XX*:*XXXX* is the rightmost 24 bits of the corresponding unicast or anycast address of the node. (Neighbor solicitation messages are sent on a local link when a node wants to determine the link-layer address of another node on the same local link, similar to Address Resolution Protocol [ARP] in IPv4.)
- **FF05::101** — All Network Time Protocol (NTP) servers in the site (site-local scope)

The site-local multicast scope has an administratively assigned radius and has no direct correlation to the (now deprecated) site-local unicast prefix of FEC0::/10.

Addresses That Are Not Unique

In very rare cases, the rightmost 24 bits of the unicast address of the target will not be unique on the link. Solicited–node multicast addresses are used in IPv6 for address resolution of an IPv6 address to a MAC address on a LAN segment. For example, consider two nodes with addresses 2001:DB8:200:300:400:500:aaaa:bbbb and 2001:DB8:200:300:400:501:aaaa:bbbb, where the link prefix is 2001:DB8:200:300::/64. These two nodes, then, would be listening to the same solicited-node multicast address. Each would receive the multicast packet, but only the node whose full address matched the full target address of the multicast packet (embedded in the data field of the multicast packet) would respond with a neighbor advertisement (which includes the actual MAC address). The other node would receive the multicast packet, but upon inspection of the embedded target address would realize that it was not the intended recipient of the request, and would not respond.

The following example describes how this situation would work.

Node A has this characteristic:

- Address 2001:DB8:200:300:400:500:1234:5678

Node B has these characteristics:

- Address 2001:DB8:200:300:500:AAAA:BBBB
- Solicited-node multicast address FF02:0:0:0:0:1:FFAA:BBBB (the same as node C)

Node C has these characteristics:

- Address 2001:DB8:200:300:501:AAAA:BBBB
- Solicited-node multicast address FF02:0:0:0:0:1:FFAA:BBBB (the same as node B)

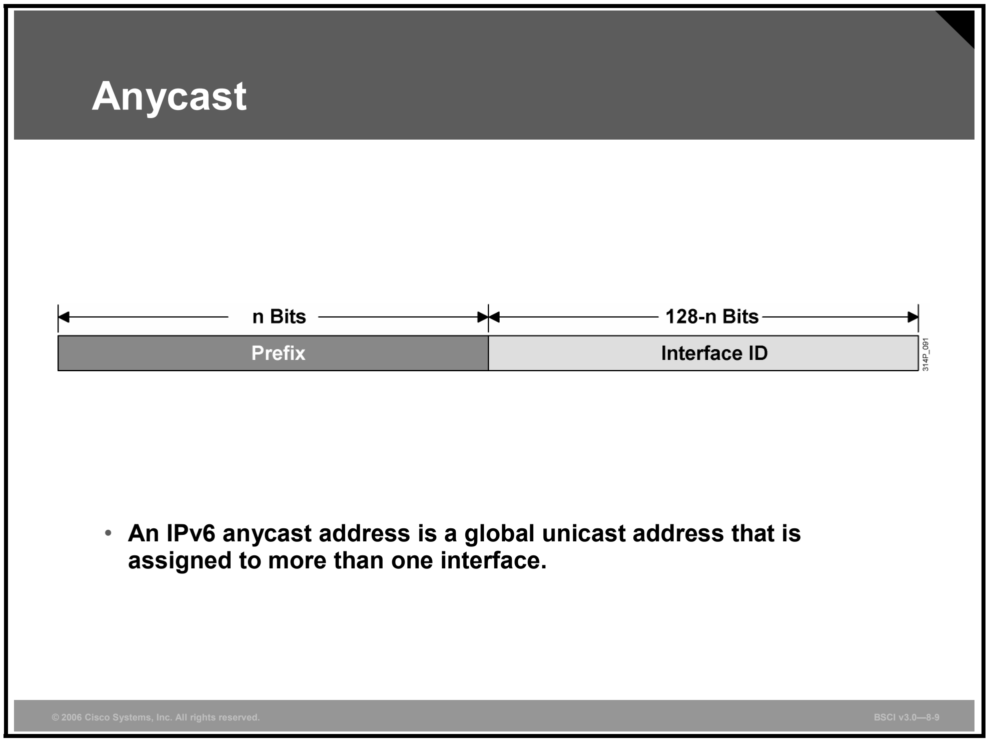
1. Node A desires to exchange packets with node B. Node A sends a neighbor discovery packet to the solicited-node multicast address of B, FF02:0:0:0:0:1:AAAA:BBBB, as described previously. Inside the packet, in addition to other data, is the full IPv6 address that node A is looking for—2001:DB8:200:300:500:AAAA:BBBB. This is called the target address.

2. Both node B and node C are listening to the same multicast address, so they both receive and process the packet.

3. Node B sees that the target address inside the packet is its own and responds as described previously.

4. Node C sees that the target address inside the packet is not its own and does not respond at all.

In this manner, nodes can have the same solicited-node multicast address on the link without causing neighbor discovery, neighbor solicitation, or neighbor advertisement to malfunction.



Anycast

n Bits

128-n Bits

Prefix

Interface ID

314P_091

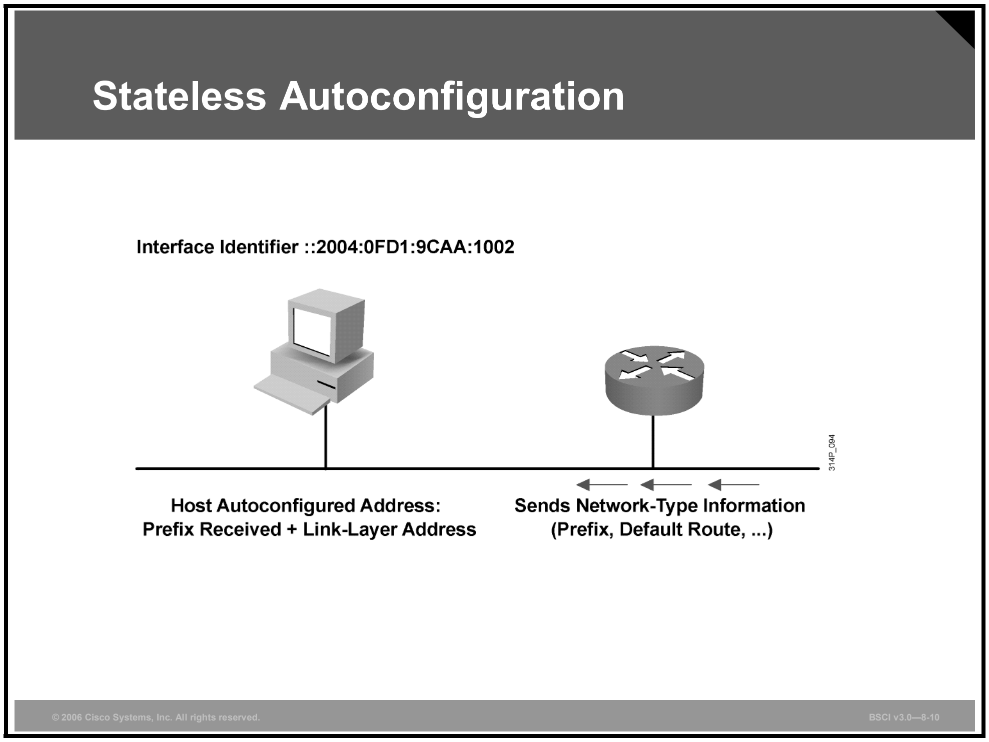
- **An IPv6 anycast address is a global unicast address that is assigned to more than one interface.**

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—8-9

An IPv6 anycast address is a global unicast address that is assigned to more than one interface. When a packet is sent to an anycast address, it is routed to the "nearest" interface having that address. In a WAN scope, the nearest interface is found according to the measure of distance of the routing protocol. In a LAN scope, the nearest interface is found according to the first neighbor that is learned about. The following describes the characteristics of the anycast:

- Anycast addresses are allocated from the unicast address space, so they are indistinguishable from the unicast address. When assigned to a node interface, the node must be explicitly configured to know that the address is an anycast address.
- 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, which enables a type of discovery mechanism to the nearest point.
- There is little experience with widespread anycast usage. A few anycast addresses are currently assigned: the router-subnet anycast and the Mobile IPv6 home agent anycast.
- An anycast address must not be used as the source address of an IPv6 packet.



Stateless Autoconfiguration

Interface Identifier ::2004:0FD1:9CAA:1002

Host Autoconfigured Address:
Prefix Received + Link-Layer Address

Sends Network-Type Information
(Prefix, Default Route, ...)

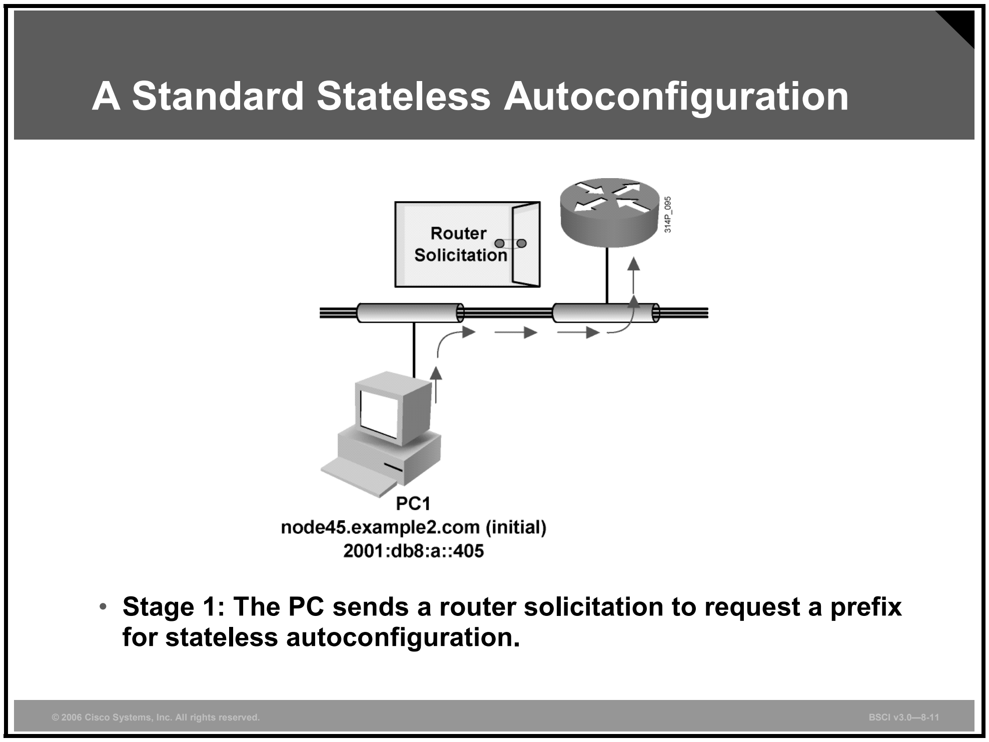
BSCI v3.0—8-10

A router on the local link can send network information, such as a 64-bit prefix of the local link network and the default route. It sends this to all the nodes on the local link. A host can autoconfigure itself by appending its IPv6 interface identifier (64-bit format) to the local link prefix (64 bits). This process results in a full 128-bit address that is usable and guaranteed to be globally unique.

A process called duplicate address translation is enabled to detect and avoid duplicate addresses.

Autoconfiguration enables the plug-and-play feature, which allows devices to connect themselves to the network without any configuration and without any servers (like DHCP servers). This key feature enables deployment of new devices on the Internet, such as cellular phones, wireless devices, home appliances, and home networks.

Note Stateless DHCP is a new concept (February 2004) that strikes a middle ground between stateless autoconfiguration and the thick-client approach of stateful DHCP. Stateless DHCP for IPv6 is also called “DHCP-lite.” See RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*.

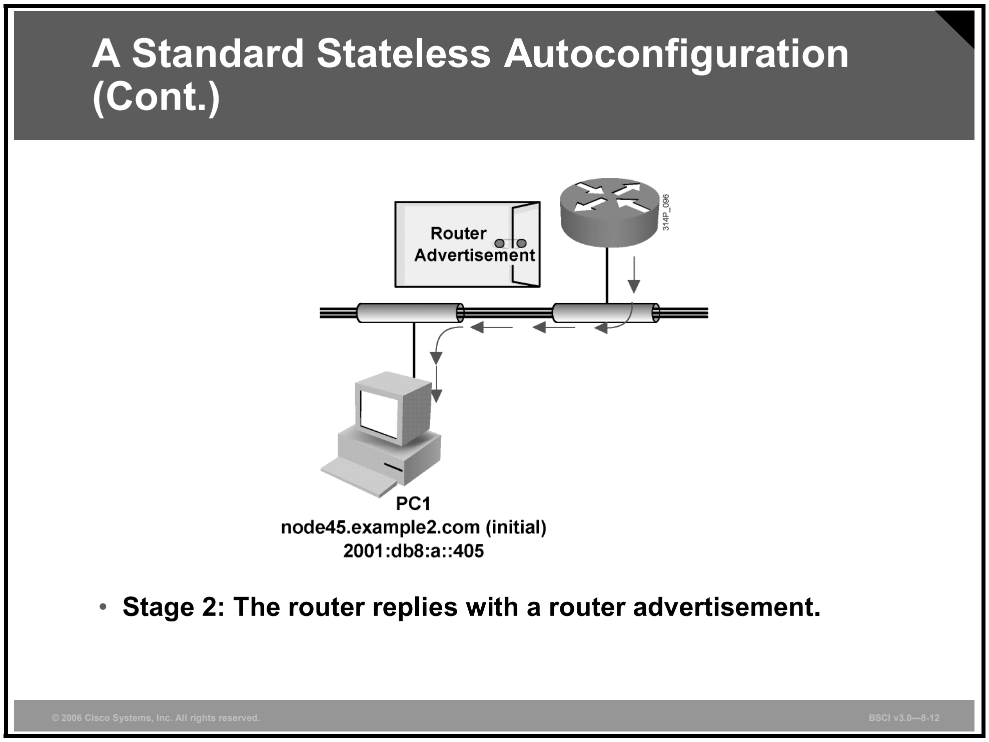


A Standard Stateless Autoconfiguration

- **Stage 1: The PC sends a router solicitation to request a prefix for stateless autoconfiguration.**

BSCI v3.0—8-11

The PC first configures its link-local address and then sends a router solicitation to request a prefix for stateless autoconfiguration.



A Standard Stateless Autoconfiguration (Cont.)

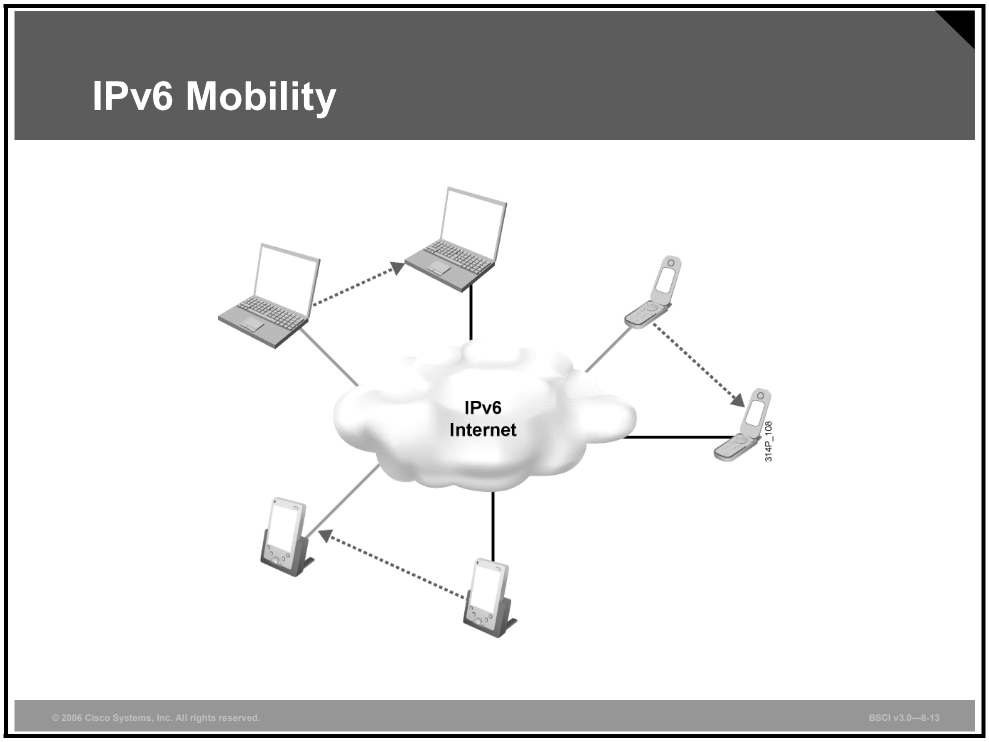
- **Stage 2: The router replies with a router advertisement.**

BSCI v3.0—8-12

The router replies with a router advertisement, including prefix information.

IPv6 Mobility

This topic describes how IPv6 simplifies mobile IP connections.



IPv6 Mobility

BSCI v3.0—8-13

Mobility is a very important feature in networks today. Mobile IP is an Internet Engineering Task Force (IETF) standard available for both IPv4 and IPv6. Mobile IP enables mobile devices to move without breaking current connections. In IPv6, mobility is built in, which means that any IPv6 node can use it as needed. However, in IPv4, mobility is a new function that must be added.

The routing headers of IPv6 make Mobile IPv6 much more efficient for end nodes than Mobile IPv4. Mobility takes advantage of the flexibility of IPv6. For example, binding uses some header options (destination) that are mandatory for every IPv6 device. Also, IPv6 mobility creates a new “mobility” extension header.

Mobile IPv6 Model

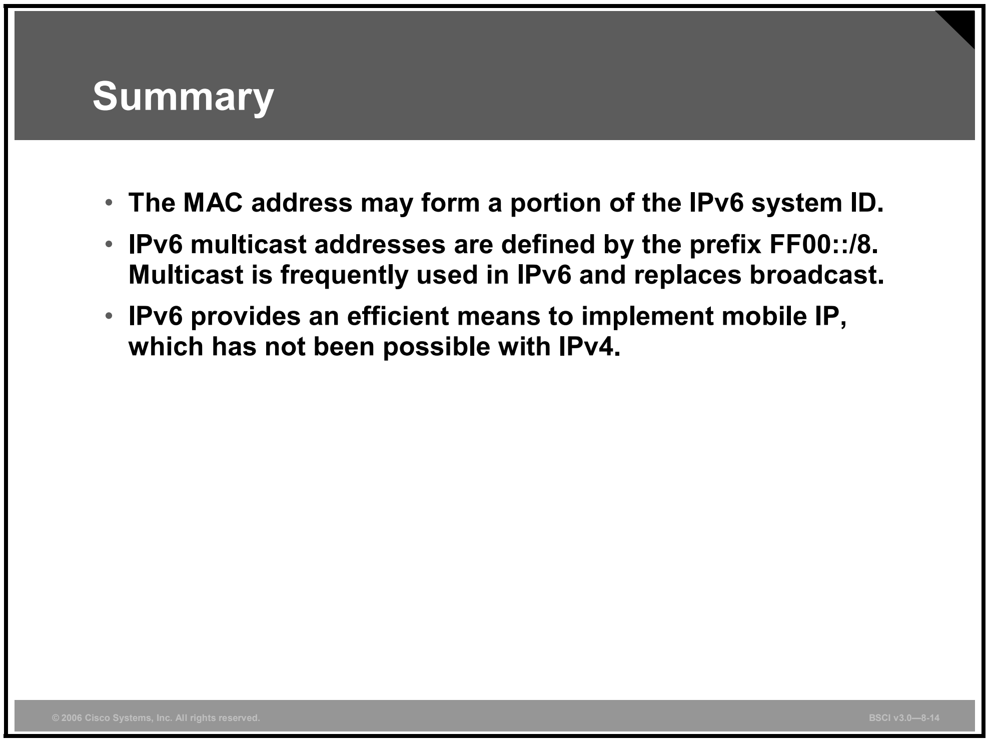
IPv6 mobility is different from IPv4 mobility in several ways:

- The IPv6 address space enables Mobile IP deployment in any kind of large environment.
- Because of the vast IPv6 address space, foreign agents are no longer required. Infrastructures do not need an upgrade to accept Mobile IPv6 nodes, so the care-of address (CoA) can be a global IPv6 routable address for all mobile nodes.
- The Mobile IPv6 model takes advantage of some of the benefits of the IPv6 protocol itself. Examples include option headers, neighbor discovery, and autoconfiguration.

- In many cases, triangle routing 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. Support is also integrated into Mobile IPv6 for allowing route optimization to coexist efficiently with routers that perform ingress filtering. Mobile IPv6 route optimization can operate securely even without prearranged security associations. It is expected that route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes.
- Mobile nodes work transparently even with other nodes that do not support mobility (same as in IPv4 mobility).
- The dynamic home agent address-discovery mechanism in Mobile IPv6 returns a single reply to the mobile node. The directed broadcast approach used in IPv4 returns separate replies from each home agent.
- Most packets sent to a mobile node while it is away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.

Summary

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



Summary

- The MAC address may form a portion of the IPv6 system ID.
- IPv6 multicast addresses are defined by the prefix FF00::/8. Multicast is frequently used in IPv6 and replaces broadcast.
- IPv6 provides an efficient means to implement mobile IP, which has not been possible with IPv4.

BSCI v3.0—8-14

Lesson 4

Using IPv6 with OSPF and Other Routing Protocols

Overview

Implementing Open Shortest Path First Protocol (OSPF) for IP version 6 (IPv6) expands on OSPF to provide support for IPv6 routing prefixes. OSPF for IPv6 is described in RFC 2740. This lesson describes the concepts and tasks you need to understand to implement OSPF for IPv6 on your network.

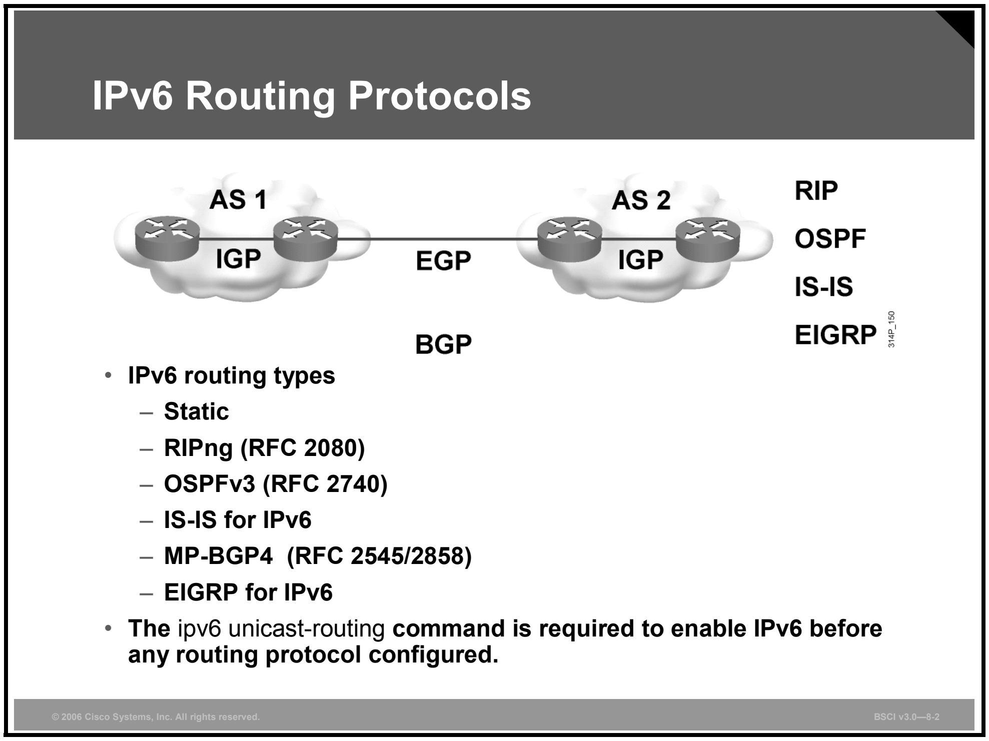
Objectives

Upon completing this topic, you will be able to identify how you use IPv6 with OSPF. This ability includes being able to meet these objectives:

- Describe the modifications that need to be made to existing routing protocols in IPv6 networks
- Describe how OSPF for IPv6 works
- Explain the similarities and differences between OSPF for IPv6 to OSPFv2
- Describe the differences between OSPF LSA types used with IPv4 and IPv6
- Explain the configuration modes and Cisco IOS attributes specific to OSPFv3
- Explain how to configure OSPFv3
- Explain how to verify OSPFv3

Describing IPv6 Routing

This topic describes the modifications that need to be made to existing routing protocols in IPv6 networks.



IPv6 Routing Protocols

- **IPv6 routing types**
  - **Static**
  - **RIPng (RFC 2080)**
  - **OSPFv3 (RFC 2740)**
  - **IS-IS for IPv6**
  - **MP-BGP4 (RFC 2545/2858)**
  - **EIGRP for IPv6**
- **The** ipv6 unicast-routing **command is required to enable IPv6 before any routing protocol configured.**

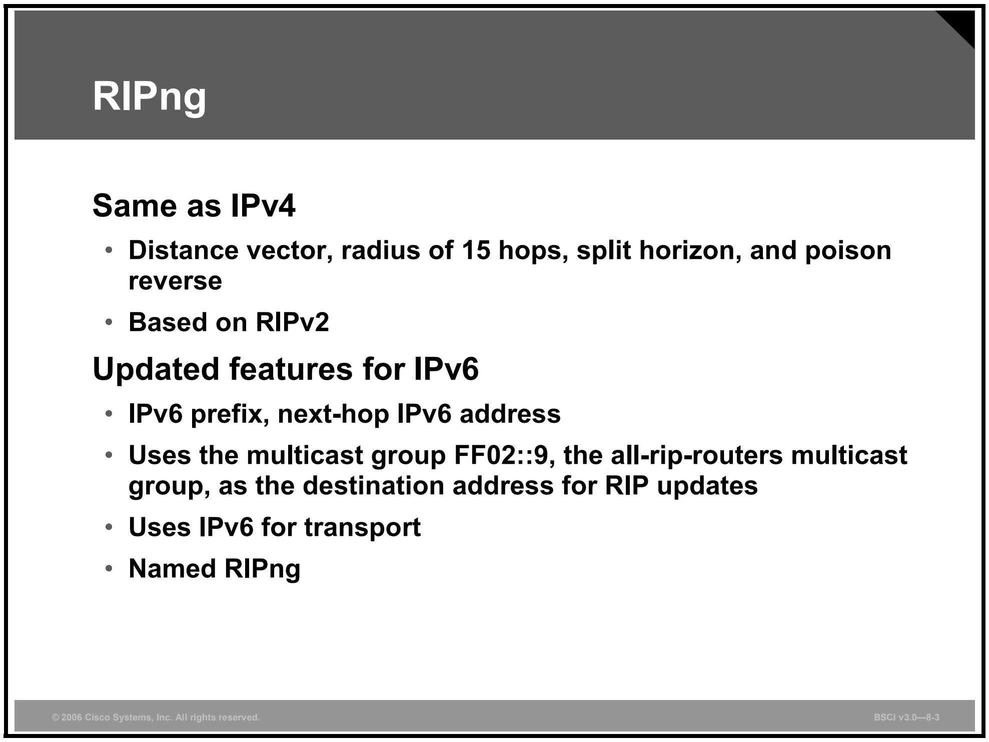
BSCI v3.0—8-2

As does IP version 4 (IPv4) classless interdomain routing (CIDR), IPv6 uses longest-prefix match routing. Recent protocol versions handle longer IPv6 addresses and different header structures. Currently, the updated routing protocols shown in the figure are available.

Static routing with IPv6 is used and configured in the same way as IPv4. There is an IPv6-specific requirement per RFC 2461: A router must be able to determine the link-local address of each of its neighboring routers to ensure that the target address of a redirect message identifies the neighbor router by its link-local address.

This requirement basically means that using a global unicast address as a next-hop address with routing is not recommended.

The Cisco IOS global command for IPv6 is **ipv6 unicast-routing.**



RIPng

Same as IPv4

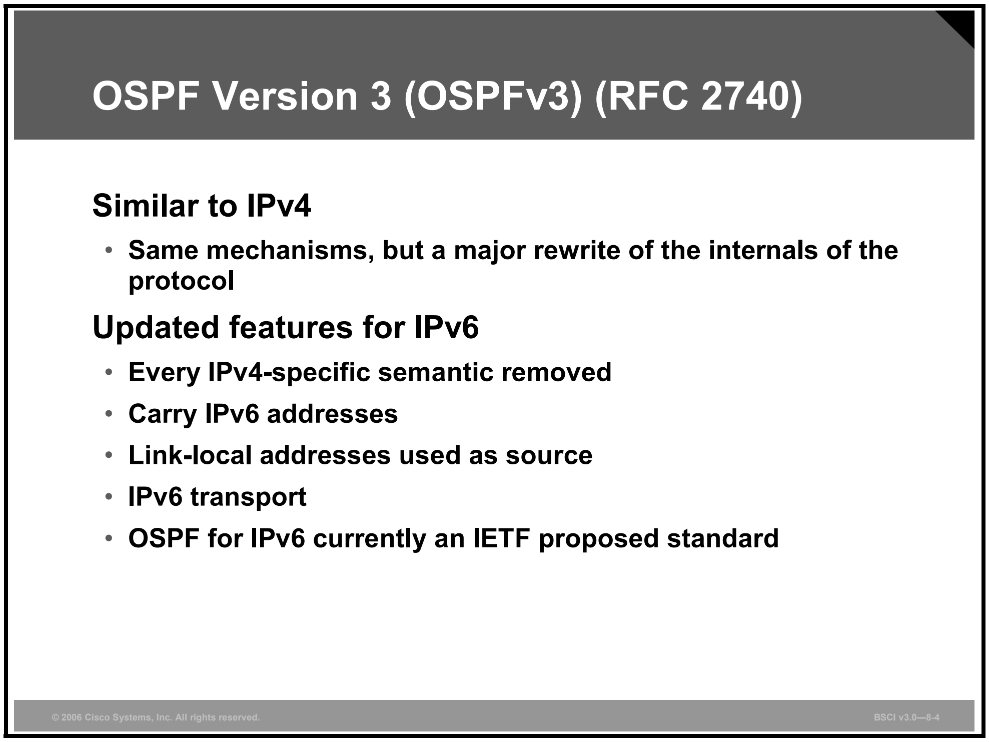
- **Distance vector, radius of 15 hops, split horizon, and poison reverse**
- **Based on RIPv2**

Updated features for IPv6

- **IPv6 prefix, next-hop IPv6 address**
- **Uses the multicast group FF02::9, the all-rip-routers multicast group, as the destination address for RIP updates**
- **Uses IPv6 for transport**
- **Named RIPng**

Routing Information Protocol next generation (RIPng, RFC 2080) is a distance vector routing protocol with a limit of 15 hops that uses split-horizon and poison reverse to prevent routing loops. IPv6 update features include these:

- Based on IPv4 RIP version 2 (RIPv2) and similar to RIPv2
- Uses IPv6 for transport
- IPv6 prefix, next-hop IPv6 address
- Uses the multicast group FF02::9, the all-RIP-routers multicast group, as the destination address for RIP updates
- Updates sent on User Datagram Protocol (UDP) port 521



OSPF Version 3 (OSPFv3) (RFC 2740)

Similar to IPv4

- **Same mechanisms, but a major rewrite of the internals of the protocol**

Updated features for IPv6

- **Every IPv4-specific semantic removed**
- **Carry IPv6 addresses**
- **Link-local addresses used as source**
- **IPv6 transport**
- **OSPF for IPv6 currently an IETF proposed standard**

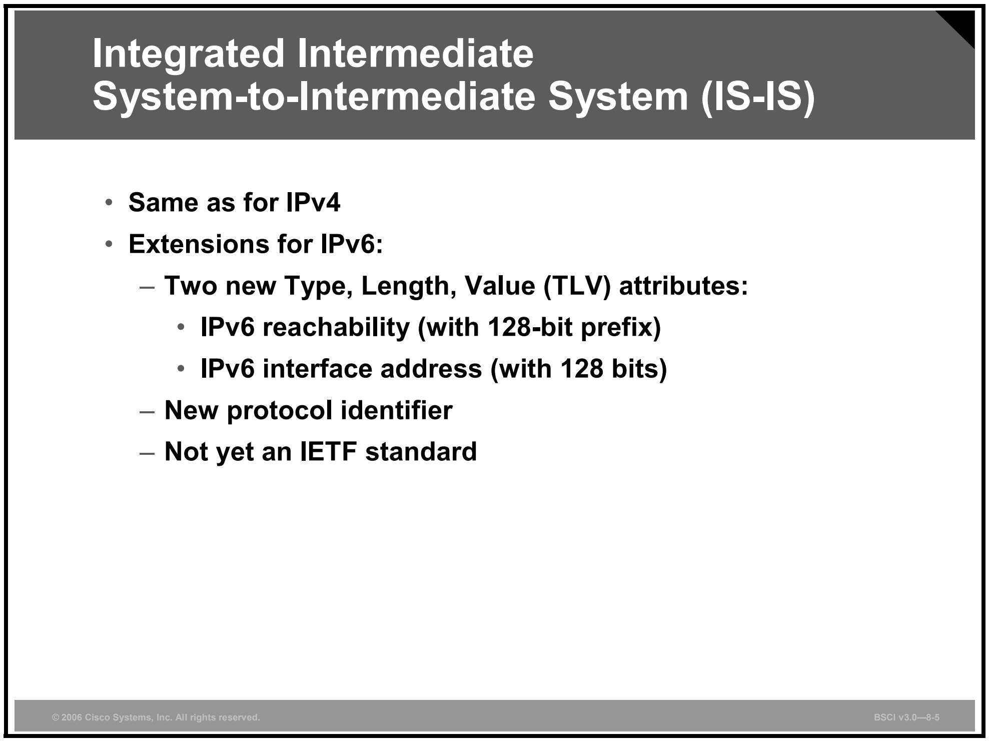
BSCI v3.0—8-4

The protocol implementation for IPv6 includes these characteristics:

- Based on OSPF version 2 (OSPFv2), with enhancements
- Distributes IPv6 prefixes
- Runs directly over IPv6
- Operates as “ships in the night” with OSPFv2

This implementation adds these IPv6-specific attributes:

- 128-bit addresses
- Link-local address
- Multiple addresses and instances per interface
- Authentication (now uses IPsec)
- OSPF version 3 (OSPFv3): Runs over a link rather than a subnet



Integrated Intermediate System-to-Intermediate System (IS-IS)

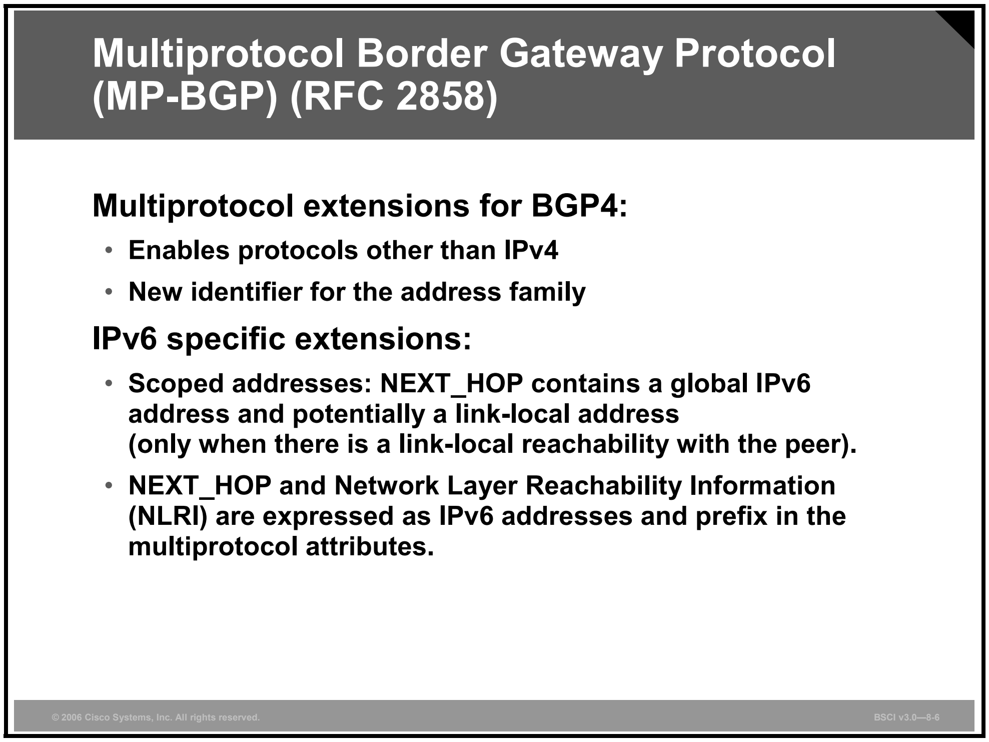
- Same as for IPv4
- Extensions for IPv6:
  - Two new Type, Length, Value (TLV) attributes:
    - IPv6 reachability (with 128-bit prefix)
    - IPv6 interface address (with 128 bits)
  - New protocol identifier
  - Not yet an IETF standard

BSCI v3.0—8-5

Large address support facilitates the IPv6 address family.

Intermediate System to Intermediate System (IS-IS) is the same as IPv4 with some extensions added:

- Two new Type, Length, Value (TLV) attributes
- IPv6 reachability
- IPv6 interface address
- New protocol ID



Multiprotocol Border Gateway Protocol (MP-BGP) (RFC 2858)

Multiprotocol extensions for BGP4:

- **Enables protocols other than IPv4**
- **New identifier for the address family**

IPv6 specific extensions:

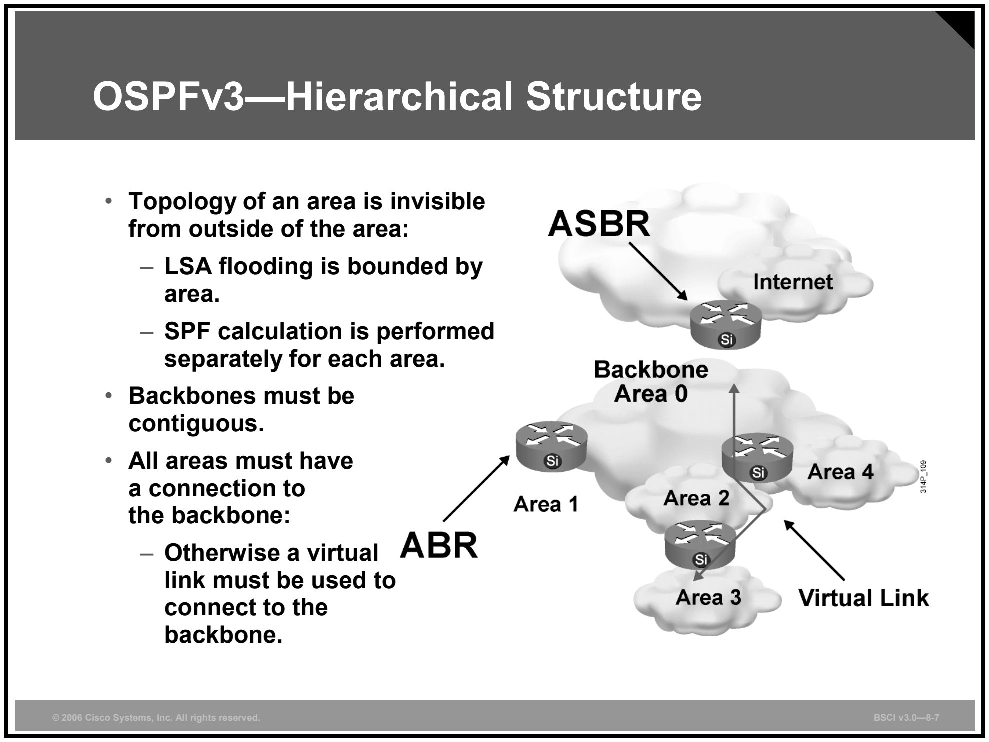
- **Scoped addresses: NEXT_HOP contains a global IPv6 address and potentially a link-local address (only when there is a link-local reachability with the peer).**
- **NEXT_HOP and Network Layer Reachability Information (NLRI) are expressed as IPv6 addresses and prefix in the multiprotocol attributes.**

To make Border Gateway Protocol version 4 (BGP4) available for other network-layer protocols, RFC 2858 (which replaces the now-obsolete RFC 2283) defines multiprotocol extensions for BGP4.

Multiprotocol BGP is used to enable BGP4 to carry information of other protocols, for example, Multiprotocol Label Switching (MPLS) and IPv6.

OSPF and IPv6

This topic describes how OSPF for IPv6 works.



OSPFv3—Hierarchical Structure

- Topology of an area is invisible from outside of the area:
  - LSA flooding is bounded by area.
  - SPF calculation is performed separately for each area.
- Backbones must be contiguous.
- All areas must have a connection to the backbone:
  - Otherwise a virtual link must be used to connect to the backbone.

© 2006 Cisco Systems, Inc. All rights reserved.

BSCI v3.0—8-7

How OSPF for IPv6 Works

OSPF is a routing protocol for IP. It is a link-state protocol, as opposed to a distance vector protocol. Think of a link as being an interface on a networking device. A link-state protocol makes its routing decisions based on the states of the links that connect source and destination machines.

The state of a link is a description of that interface and its relationship to its neighboring networking devices. The interface information includes the IPv6 prefix of the interface, the network mask, the type of network that it is connected to, the routers connected to that network, and so on.

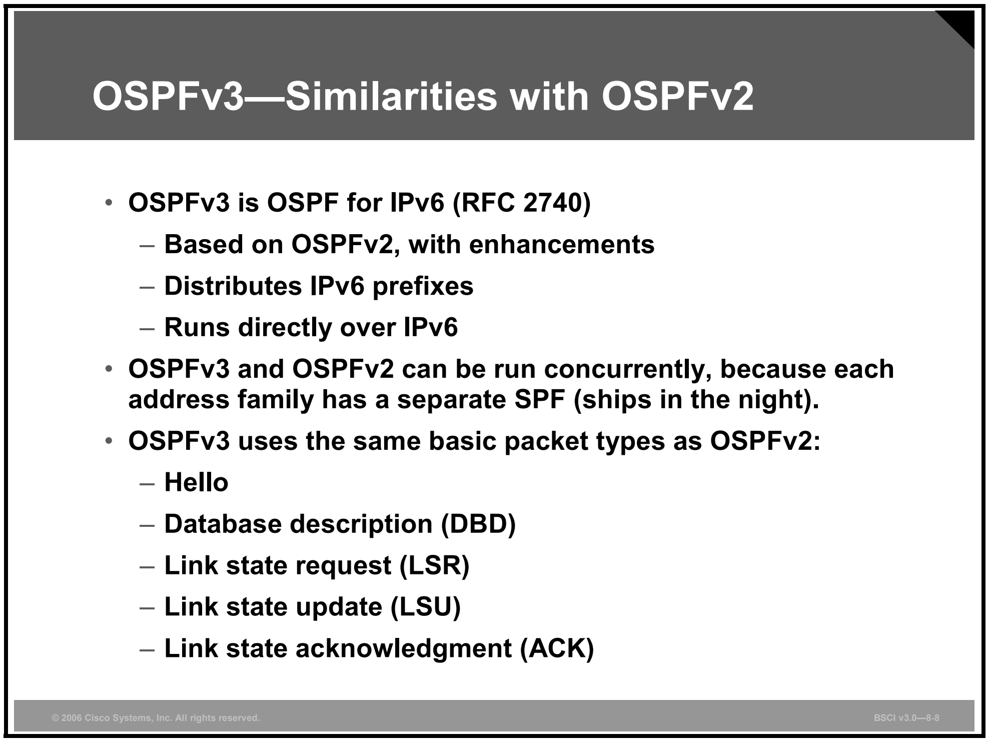
This information is propagated in various types of link-state advertisements (LSAs). A collection of LSA data on a router is stored in a link-state database (LSDB). The contents of the database, when subjected to Dijkstra's algorithm, result in the creation of the OSPF routing table.

The difference between the database and the routing table is that the database contains a complete collection of raw data; the routing table contains a list of shortest paths to known destinations via specific router interface ports.

OSPFv3, which is described in RFC 2740, supports IPv6.

Comparing OSPF for IPv6 to OSPFv2

This topic explains the similarities and differences between OSPF for IPv6 to OSPFv2.



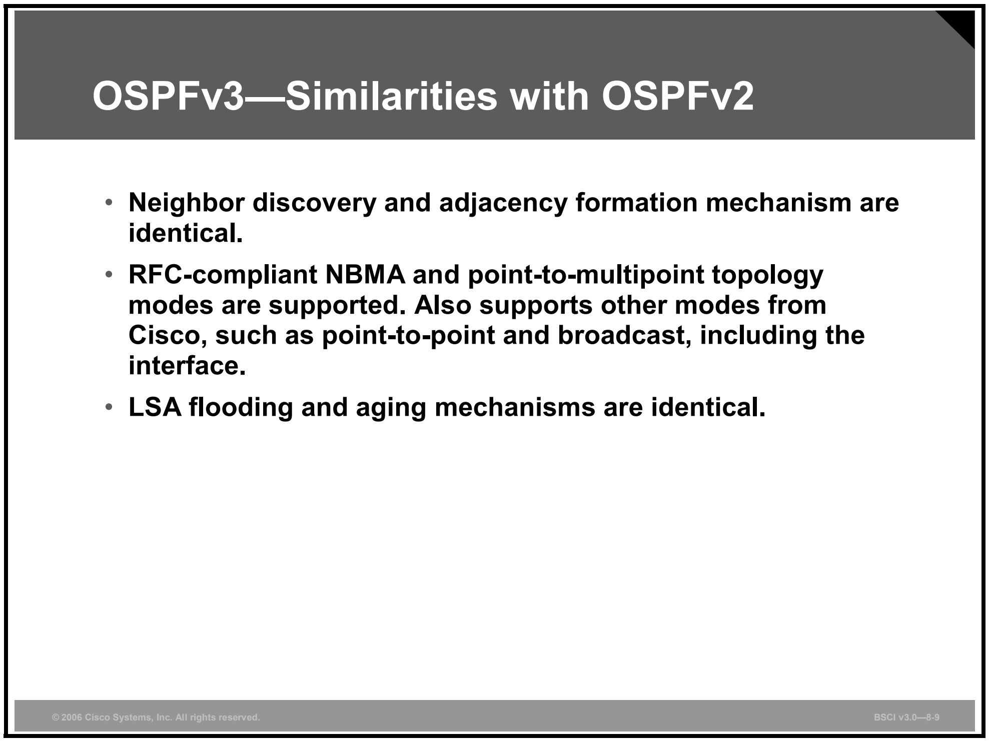
OSPFv3—Similarities with OSPFv2

- **OSPFv3 is OSPF for IPv6 (RFC 2740)**
  - **Based on OSPFv2, with enhancements**
  - **Distributes IPv6 prefixes**
  - **Runs directly over IPv6**
- **OSPFv3 and OSPFv2 can be run concurrently, because each address family has a separate SPF (ships in the night).**
- **OSPFv3 uses the same basic packet types as OSPFv2:**
  - **Hello**
  - **Database description (DBD)**
  - **Link state request (LSR)**
  - **Link state update (LSU)**
  - **Link state acknowledgment (ACK)**

Although most of the algorithms of OSPFv2 are the same as those of OSPFv3, some changes have been made in OSPFv3, particularly to handle the increased address size in IPv6 and the fact that OSPF runs directly over IP.

Other similarities to OSPFv2 include these:

- OSPFv3 uses the same basic packet types as OSPFv2, such as hello, database description (DBD, also called database description packet [DDP]), link-state request (LSR), link-state update (LSU), and LSA.
- Mechanisms for neighbor discovery and adjacency formation are identical.
- Operations of OSPFv3 over the RFC-compliant nonbroadcast multiaccess (NBMA) and point-to-multipoint topology modes are supported. OSPFv3 also supports the other modes from Cisco such as point-to-point and broadcast, including the interface.
- LSA flooding and aging are the same for OSPFv2 and OSPFv3.



OSPFv3—Similarities with OSPFv2

- **Neighbor discovery and adjacency formation mechanism are identical.**
- **RFC-compliant NBMA and point-to-multipoint topology modes are supported. Also supports other modes from Cisco, such as point-to-point and broadcast, including the interface.**
- **LSA flooding and aging mechanisms are identical.**

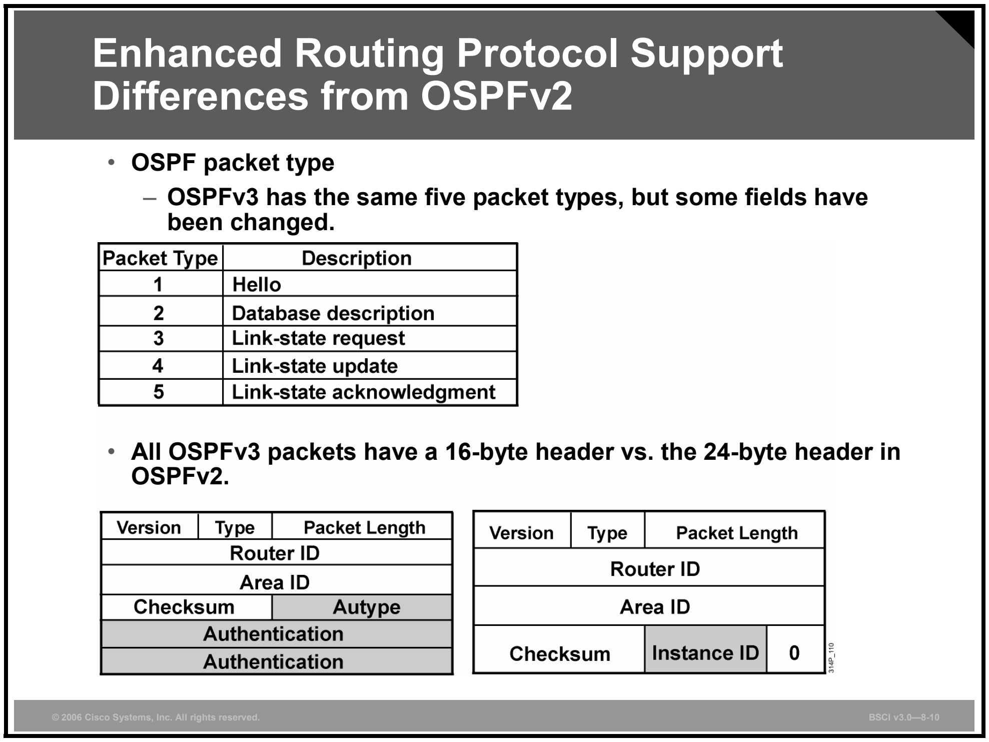
BSCI v3.0—8-9

Because OSPFv2 is heavily dependent on the IPv4 address for its operation, changes were necessary in the OSPFv3 protocol to support IPv6, as outlined in RFC 2740, *OSPF for IPv6*.

Some of the notable changes include platform-independent implementation, per-link protocol processing rather than per-node processing, explicit support for multiple instances per link, and changes in authentication and packet format.

IPv6 OSPF is now an IETF proposed standard. Like RIPng, IPv6 OSPFv3 uses IPv6 for transport and uses link-local addresses as source addresses.

All of the optional capabilities of OSPF for IPv4, including on-demand circuit support, not-so-stubby areas (NSSAs), and the extensions to Multicast OSPF (MOSPF), are also supported in OSPF for IPv6.



Enhanced Routing Protocol Support Differences from OSPFv2

- **OSPF packet type**
  - **OSPFv3 has the same five packet types, but some fields have been changed.**

Packet Type	Description
1	Hello
2	Database description
3	Link-state request
4	Link-state update
5	Link-state acknowledgment

- **All OSPFv3 packets have a 16-byte header vs. the 24-byte header in OSPFv2.**

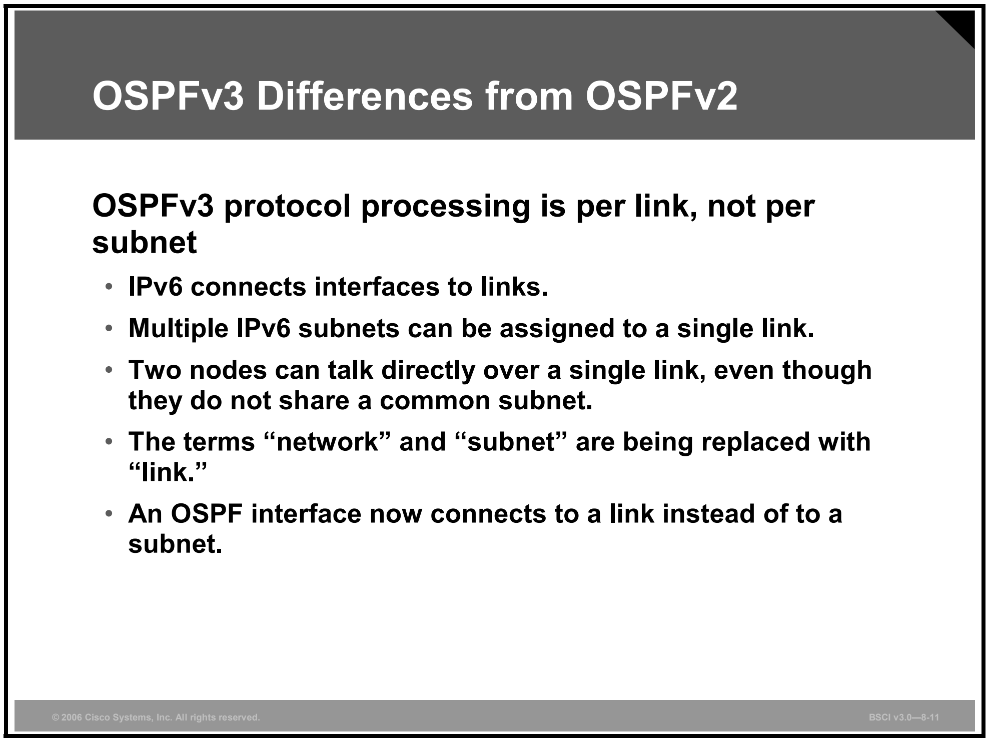
Version	Type	Packet Length
Router ID		
Area ID		
Checksum	Autype	
Authentication		
Authentication		

Version	Type	Packet Length	
Router ID			
Area ID			
Checksum	Instance ID	0	

OSPFv2 is primarily concerned with the subnet on which it is operating, but OSPFv3 is concerned with the links to which the router is connected.

OSPFv2 does not define or allow for multiple instances per link, although similar functionality could be furnished by other mechanisms, such as subinterfaces. OSPFv3 has explicit support for instances through the instance field.

Authentication is no longer part of OSPF; it is now the job of IPv6 to make sure that the right level of authentication is in use.



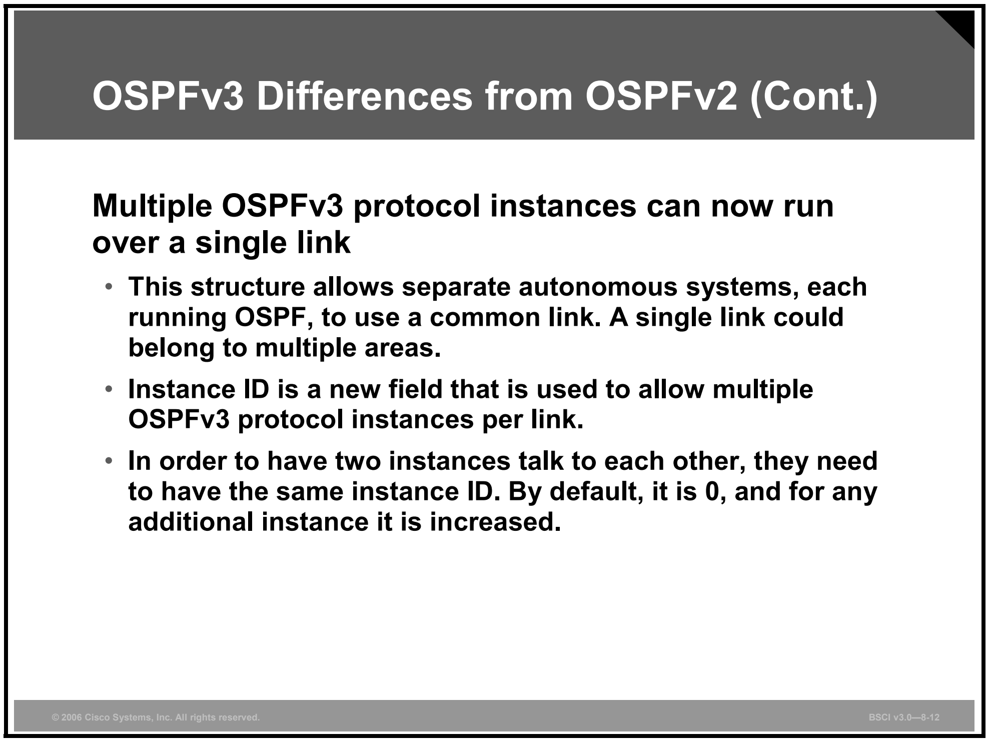
OSPFv3 Differences from OSPFv2

OSPFv3 protocol processing is per link, not per subnet

- **IPv6 connects interfaces to links.**
- **Multiple IPv6 subnets can be assigned to a single link.**
- **Two nodes can talk directly over a single link, even though they do not share a common subnet.**
- **The terms “network” and “subnet” are being replaced with “link.”**
- **An OSPF interface now connects to a link instead of to a subnet.**

These are the differences between OSPFv2 and OSPFv3:

- **OSPFv3 runs over a link:** The network statement in the router subcommand mode of OSPFv2 is replaced by an OSPFv3 command to apply to the interface configuration. It is possible to have multiple instances per link.
- **Link-local addresses are used:** OSPFv3 uses IPv6 link-local addresses to identify the OSPFv3 adjacency neighbors.



OSPFv3 Differences from OSPFv2 (Cont.)

Multiple OSPFv3 protocol instances can now run over a single link

- **This structure allows separate autonomous systems, each running OSPF, to use a common link. A single link could belong to multiple areas.**
- **Instance ID is a new field that is used to allow multiple OSPFv3 protocol instances per link.**
- **In order to have two instances talk to each other, they need to have the same instance ID. By default, it is 0, and for any additional instance it is increased.**

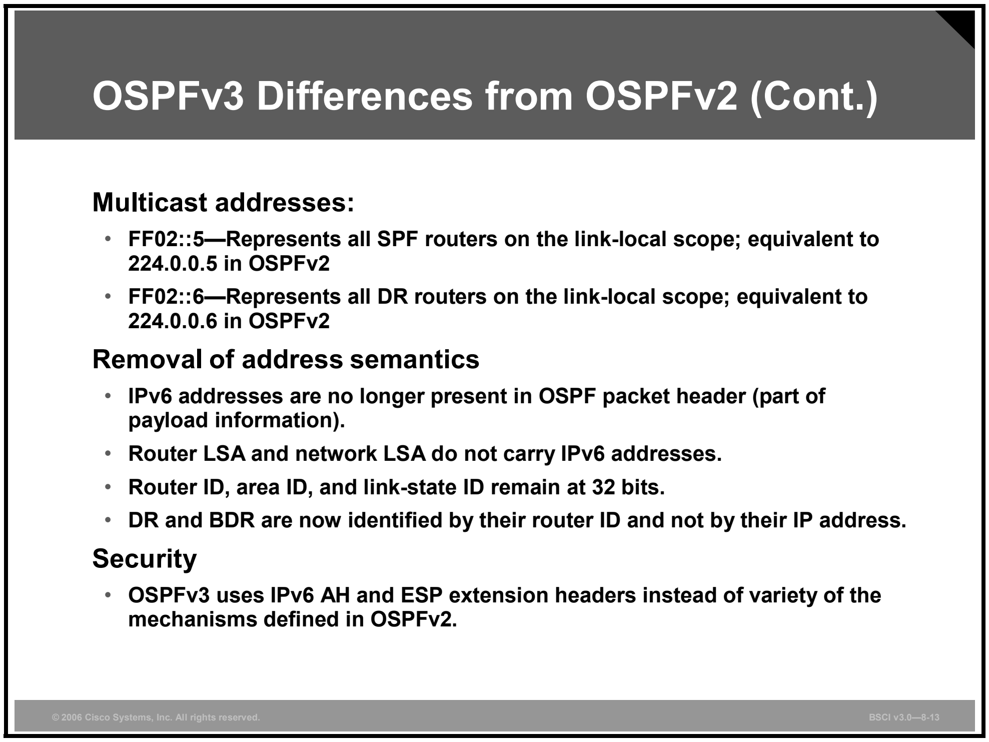
BSCI v3.0—8-12

IPv6 uses the term “link” to indicate “a communication facility or medium over which nodes can communicate at the link layer.”

Interfaces connect to links. Multiple IP subnets can be assigned to a single link, and two nodes can talk directly over a single link, even if they do not share a common IP subnet (IPv6 prefix). For this reason, OSPF for IPv6 runs per link instead of the IPv4 behavior of per IP subnet.

The terms “network” and “subnet” used in the IPv4 OSPF specification should generally be replaced by “link.” Likewise, an OSPF interface now connects to a link instead of an IP subnet, and so on.

This change affects the receiving of OSPF protocol packets and the contents of hello packets and network LSAs.



OSPFv3 Differences from OSPFv2 (Cont.)

Multicast addresses:

- **FF02::5—Represents all SPF routers on the link-local scope; equivalent to 224.0.0.5 in OSPFv2**
- **FF02::6—Represents all DR routers on the link-local scope; equivalent to 224.0.0.6 in OSPFv2**

Removal of address semantics

- **IPv6 addresses are no longer present in OSPF packet header (part of payload information).**
- **Router LSA and network LSA do not carry IPv6 addresses.**
- **Router ID, area ID, and link-state ID remain at 32 bits.**
- **DR and BDR are now identified by their router ID and not by their IP address.**

Security

- **OSPFv3 uses IPv6 AH and ESP extension headers instead of variety of the mechanisms defined in OSPFv2.**

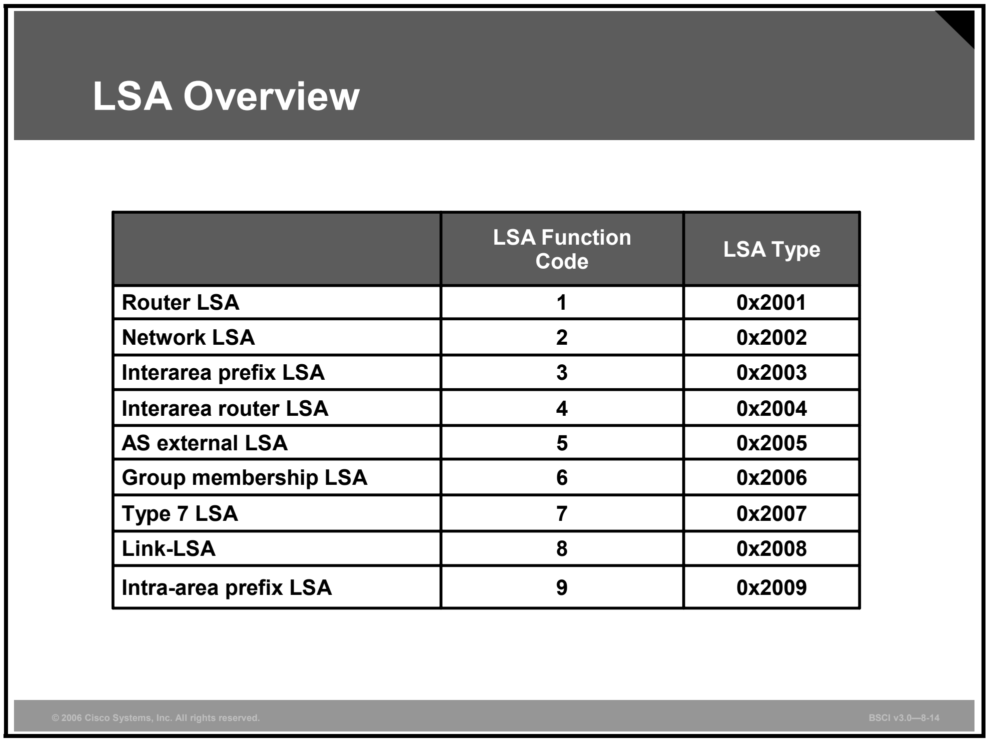
BSCI v3.0—8-13

Differences from OSPFv2 include the following:

- Multicast addresses
  - FF02::5—Represents all shortest path first (SPF) routers on the link-local scope, equivalent to 224.0.0.5 in OSPFv2
  - FF02::6—Represents all designated routers (DRs) on the link-local scope, equivalent to 224.0.0.6 in OSPFv2
- Removal of address semantics
  - IPv6 addresses are no longer present in the OSPF packet header (part of payload information).
  - Router LSAs and network LSAs do not carry IPv6 addresses.
  - The router ID, area ID and link-state ID remain at 32 bits.
  - The DR and backup designated router (BDR) are now identified by their router ID and not by their IP address.
- Security
  - OSPFv3 uses IPv6 Authentication Header (AH) and Encapsulating Security Payload (ESP) extension headers instead of the variety of mechanisms defined in OSPFv2.

LSA Types for IPv6

This topic describes the differences between OSPF LSA types used with IPv4 and IPv6.



LSA Overview

	LSA Function Code	LSA Type
Router LSA	1	0x2001
Network LSA	2	0x2002
Interarea prefix LSA	3	0x2003
Interarea router LSA	4	0x2004
AS external LSA	5	0x2005
Group membership LSA	6	0x2006
Type 7 LSA	7	0x2007
Link-LSA	8	0x2008
Intra-area prefix LSA	9	0x2009

LSAs

OSPFv3 LSA features include the following:

- The LSA is composed of a router ID, area ID, and link-state ID. They are each 32 bits and are not derived from an IPv4 address.
- Router LSAs and network LSAs contain only 32-bit IDs. They do not contain addresses.
- LSAs have flooding scopes that define a diameter that they should be flooded to:
  - **Link local:** Flood all routers on the link.
  - **Area:** Flood all routers within an OSPF area.
  - **Autonomous system (AS):** Flood all routers within the entire OSPF AS.
- OSPFv3 supports the forwarding of unknown LSAs based on the flooding scope. This can be useful in an NSSA.
- OSPFv3 now takes advantage of IPv6 multicasting, using FF02::5 for all OSPF routers and FF02::6 for the OSPF DR and the OSPF BDR.

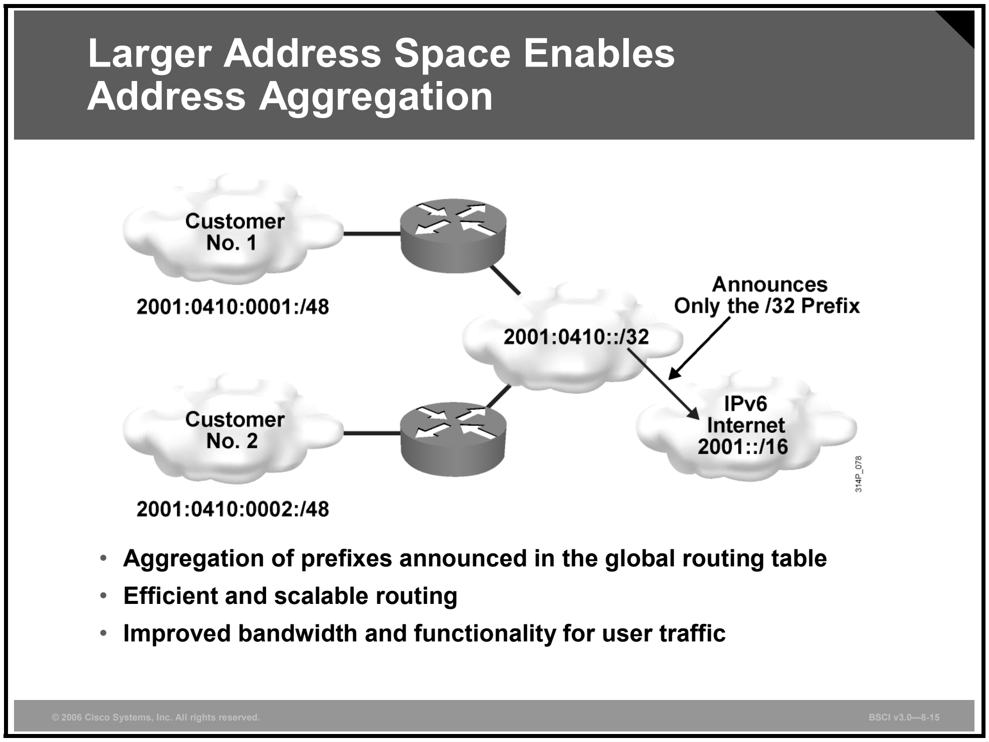
The two renamed LSAs are as follows:

- **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, 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.

The two new LSAs in IPv6 are as follows:

- **Link LSAs (type 8**)**:** Type 8 LSAs have link-local flooding scope and are never flooded beyond the link with which they are associated. Link LSAs provide the link-local address of the router to all other routers attached to the link, inform other routers attached to the link of a list of IPv6 prefixes to associate with the link, and allow the router to assert a collection of options bits to associate with the network LSA that will be originated for the link.
- **Intra-area prefix LSAs (type 9):** A router can originate multiple intra-area prefix LSAs for each router or transit network, each with a unique link-state ID. The link-state ID for each intra-area prefix LSA describes its association to either the router LSA or the network LSA. The link-state ID also contains prefixes for stub and transit networks.



Larger Address Space Enables Address Aggregation

- Aggregation of prefixes announced in the global routing table
- Efficient and scalable routing
- Improved bandwidth and functionality for user traffic

BSCI v3.0—8-15

Address Prefix

An address prefix occurs in almost all newly defined LSAs. The prefix is represented by three fields: Prefix Length, Prefix Options, and Address Prefix. In OSPF for IPv6, addresses for these LSAs are expressed as *prefix, prefix length* instead of *address, mask*.

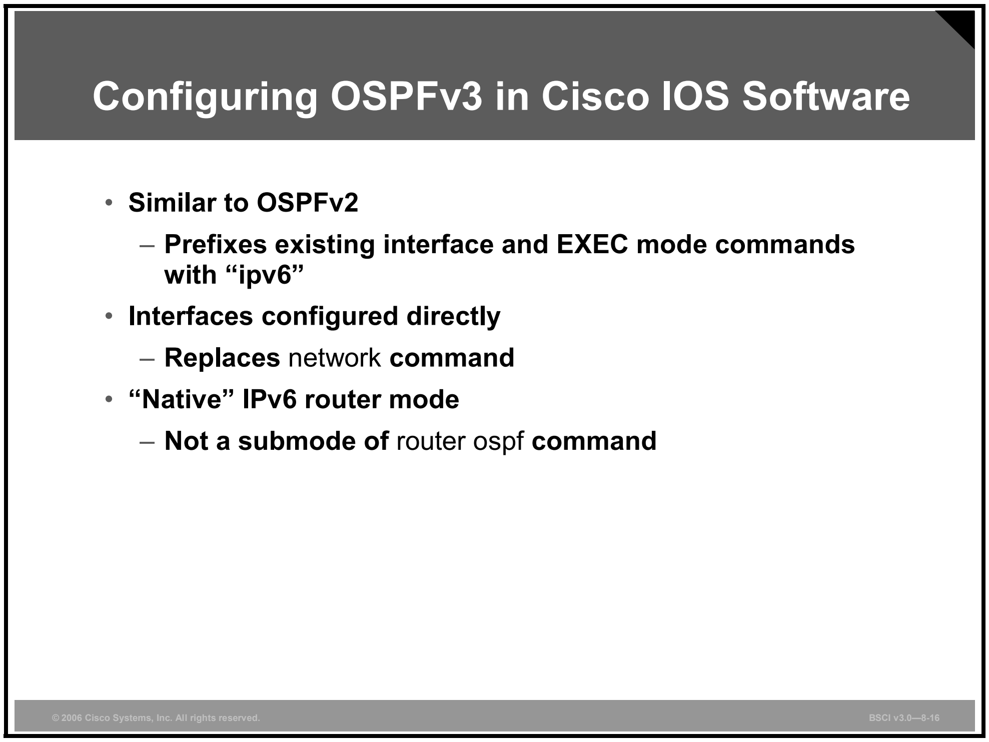
The default route is expressed as a prefix with length 0.

Type 3 and type 9 LSAs carry all IPv6 prefix information, which, in IPv4, is included in router LSAs and network LSAs.

Tip	For more information on address prefixes, refer to section 3.4.3.7, "Intra-Area-Prefix-LSAs," in RFC 2740.

Introducing OSPFv3 Configuration

This topic explains the configuration modes and Cisco IOS attributes specific to OSPFv3.



Configuring OSPFv3 in Cisco IOS Software

- **Similar to OSPFv2**
  - **Prefixes existing interface and EXEC mode commands with "ipv6"**
- **Interfaces configured directly**
  - **Replaces** network **command**
- **"Native" IPv6 router mode**
  - **Not a submode of** router ospf **command**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-16

OSPFv3 has the following Cisco IOS attributes:

- **network area command:** The way to identify IPv6 networks that are part of the OSPFv3 network is different from OSPFv2 configuration. The **network area** command in OSPFv2 is replaced by a configuration in which interfaces are directly configured to specify that IPv6 networks are part of the OSPFv3 network.
- **Native IPv6 router mode:** The configuration of OSPFv3 is not a subcommand mode of the **router ospf** command (as it is in OSPFv2 configuration).

To configure OSPFv3, first enable IPv6, and then enable OSPFv3 and specify a router ID, using the following commands:

```
Router(config)#ipv6 unicast-routing

Router(config)#ipv6 router ospf process-id

Router(config-rtr)#router-id router-id
```

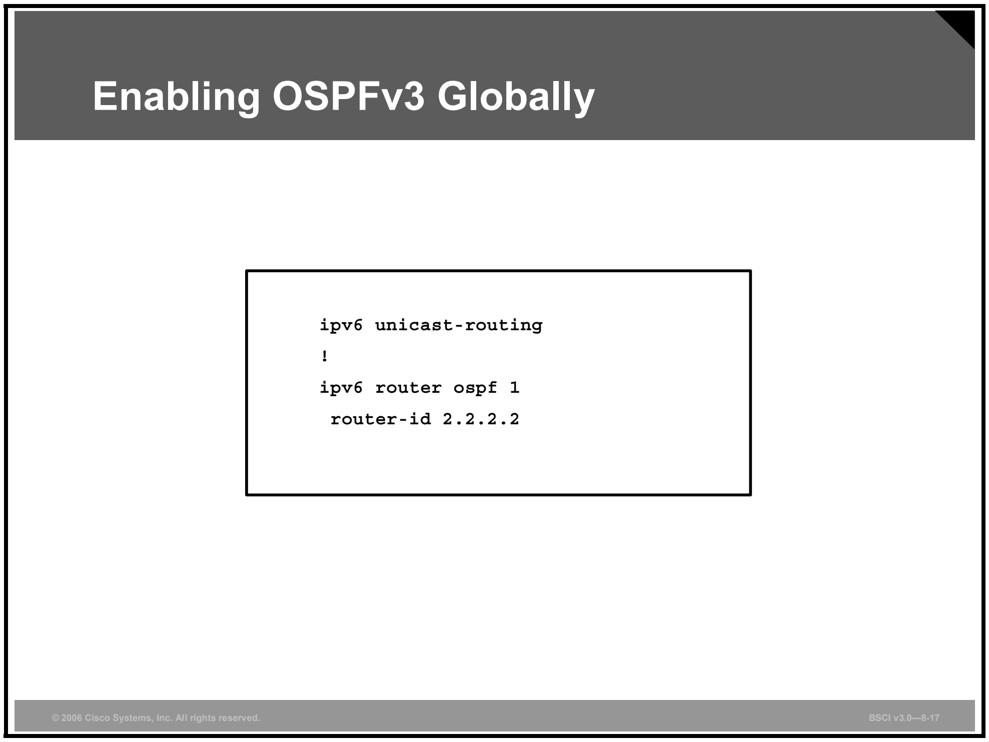
The table provides IPv6 and OSPFv3 commands, command examples, and descriptions.

IPv6 and OSPFv3 Commands, Command Examples, and Descriptions

Command	Description
Router(config)#**ipv6 unicast-routing**	Enables the forwarding of IPv6 unicast datagrams.
Router(config)#**ipv6 router ospf** *process-id*	Enables an OSPF process on the router. The *process-id* parameter identifies a unique OSPFv3 process. This command is used on a global basis.
Example	**Description**
Router(config)#**ipv6 router ospf** *100*	Enables the OSPFv3 process number 100 on the router.
Router (config-rtr)# **router-id** *router-id*	For an IPv6-only router, a router ID parameter must be defined in the OSPFv3 configuration as an IPv4 address using the **router-id** *router-id* command. You can use any IPv4 address as the router ID value.
Example	**Description**
Router (config-rtr)# **router-id** *192.168.1.12*	Identifies 192.168.1.12 as the router ID for this router. The router ID must be unique on each router.

Configuring OSPFv3

This topic explains how to configure OSPFv3.



Enabling OSPFv3 Globally

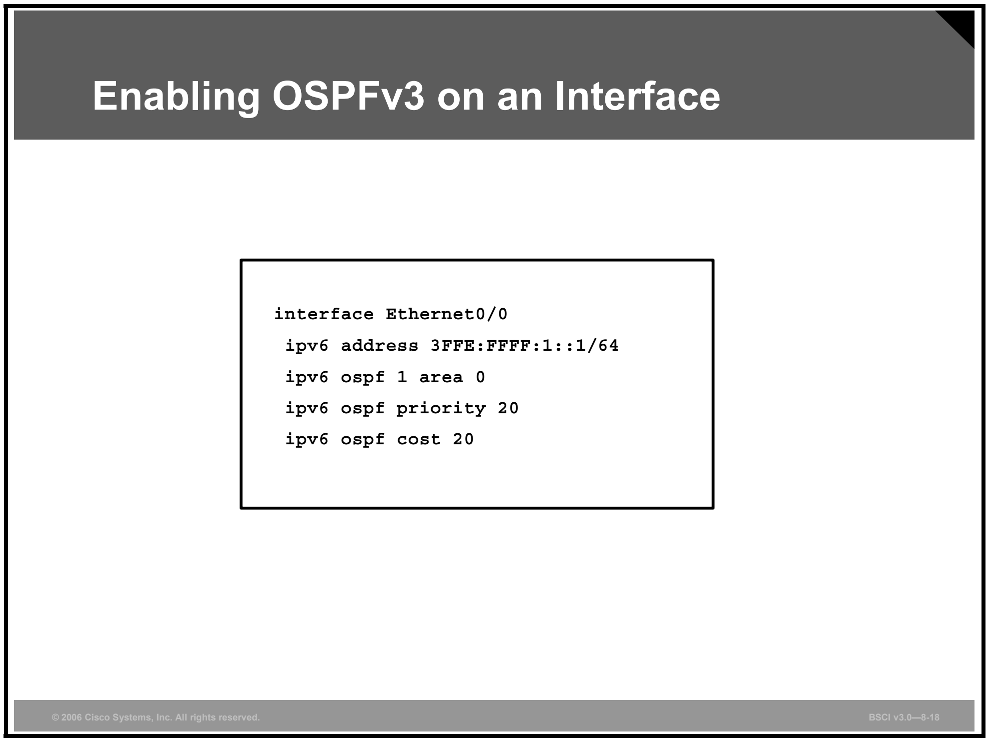
```
ipv6 unicast-routing
!
ipv6 router ospf 1
 router-id 2.2.2.2
```

BSCI v3.0—8-17

The table provides the detailed steps for configuring OSPFv3; an example is shown in the figure.

Details for Configuring OSPFv3

Command	Description
Router(config)#**ipv6 router ospf** *process-id*	Enables an OSPF process on the router. The process ID parameter identifies a unique OSPFv3 process. This command is used on a global basis.
Example	**Description**
Router(config)#**ipv6 router ospf** *1*	Enables the OSPFv3 process number 1 on the router.
Router (config-router)# **router-id** *router-id*	For an IPv6-only router, a router ID parameter must be defined in the OSPFv3 configuration as an IPv4 address using the **router-id** *router-id* command. You can use any IPv4 address as the router ID value.
Example	**Description**
Router (config-router)# **router-id 2.2.2.2**	Identifies 2.2.2.2 as the router ID for this router. The router ID must be unique on each router.



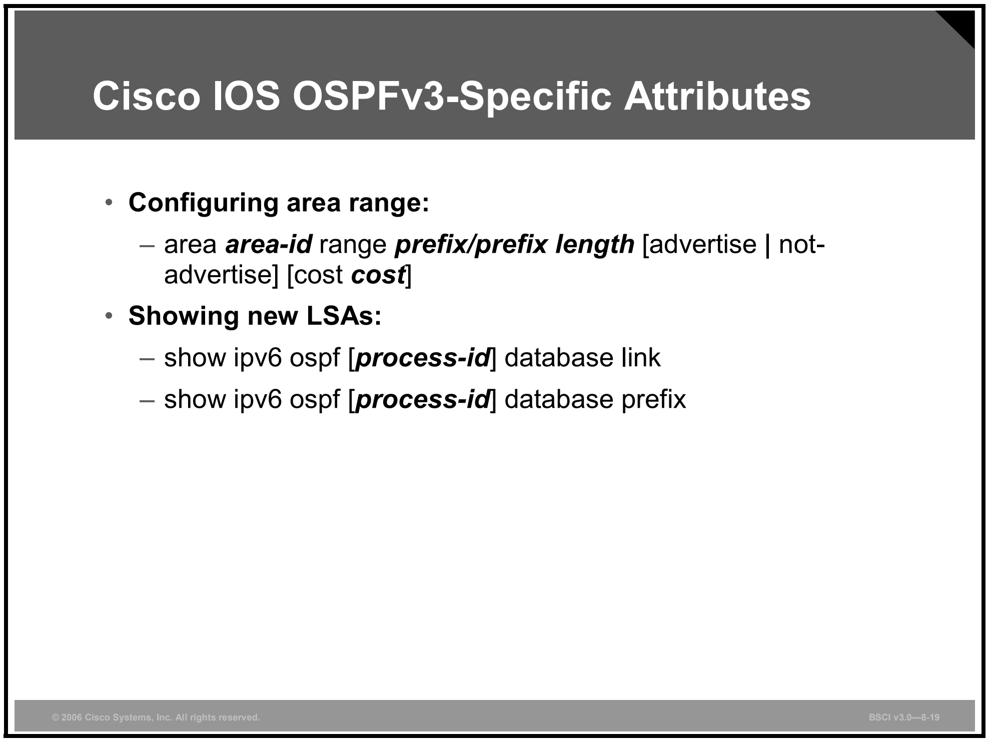
Enabling OSPFv3 on an Interface

```
interface Ethernet0/0
 ipv6 address 3FFE:FFFF:1::1/64
 ipv6 ospf 1 area 0
 ipv6 ospf priority 20
 ipv6 ospf cost 20
```

The table provides the detailed steps for enabling IPv6 and OSPFv3 on an interface; an example is shown in the figure.

Detailed Steps for Enabling IPv6 and OSPFv3 on an Interface

Step	Command or Action	Purpose
1.	Router>**enable**	Enables privileged EXEC mode. Enter your password if prompted.
2.	Router#**configure terminal**	Enters global configuration mode.
3.	Router(config)#**interface** *type number*	Specifies an interface type and number, and places the router in interface configuration mode.
4.	Router(config-if)#**ipv6 address** *address/prefix-length* [**eui-64**]	Configures an IPv6 address for an interface and enables IPv6 processing on the interface. The **eui-64** parameter forces the router to complete the address low-order 64-bits by using an EUI-64 interface ID.
5.	Router(config-if)#**ipv6 ospf** *process-id* **area** *area-id* [**instance** *instance-id*]	Enables OSPF for IPv6 on an interface.
6.	Router(config-if)#**router ospf priority** *priority number*	The priority number is used to in the designated router election.
7.	Router(config-if)#**router ospf cost** *cost*	The cost of sending a packet on the interface, expressed in the link state metric.



Cisco IOS OSPFv3-Specific Attributes

- **Configuring area range:**
  - area ***area-id*** range ***prefix/prefix length*** [advertise | not-advertise] [cost ***cost***]
- **Showing new LSAs:**
  - show ipv6 ospf [***process-id***] database link
  - show ipv6 ospf [***process-id***] database prefix

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-19

Defining an OSPF IPv6 Area Range

To consolidate and summarize routes at an area boundary use the **area** *area-id* **range** *ipv6-prefix/prefix-length* [**advertise | not-advertise**] [**cost** *cost*] IPv6 OSPF router command. This is an example:

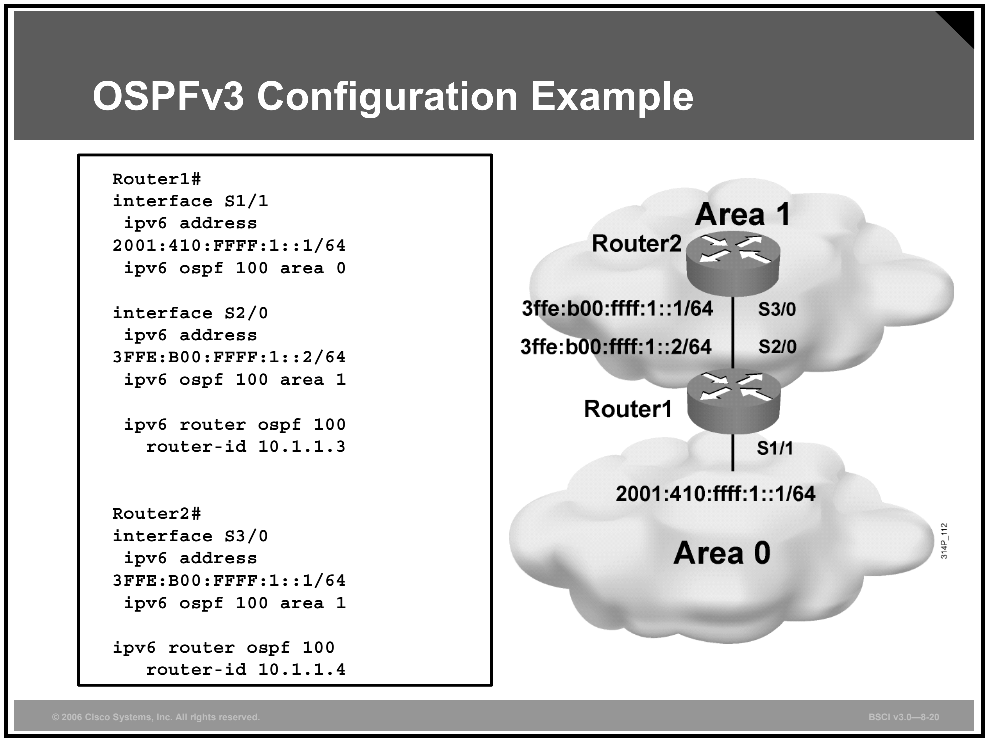
```
Router(config)#ipv6 router ospf 1
Router(config-rtr)#area range 1 2001:0DB8::/48
```

The cost of the summarized routes will be the highest cost of the routes being summarized. For example, if the following routes are summarized:

```
OI 2001:0DB8:0:0:7::/64 [110/20]
via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2001:0DB8:0:0:8::/64 [110/100]
via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
OI 2001:0DB8:0:0:9::/64 [110/20]
via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
```

They become one summarized route:

```
OI 2001:0DB8::/48 [110/100]
via FE80::A8BB:CCFF:FE00:6F00, Ethernet0/0
```



OSPFv3 Configuration Example

```
Router1#
interface S1/1
 ipv6 address
2001:410:FFFF:1::1/64
 ipv6 ospf 100 area 0

interface S2/0
 ipv6 address
3FFE:B00:FFFF:1::2/64
 ipv6 ospf 100 area 1

 ipv6 router ospf 100
   router-id 10.1.1.3


Router2#
interface S3/0
 ipv6 address
3FFE:B00:FFFF:1::1/64
 ipv6 ospf 100 area 1

ipv6 router ospf 100
   router-id 10.1.1.4
```

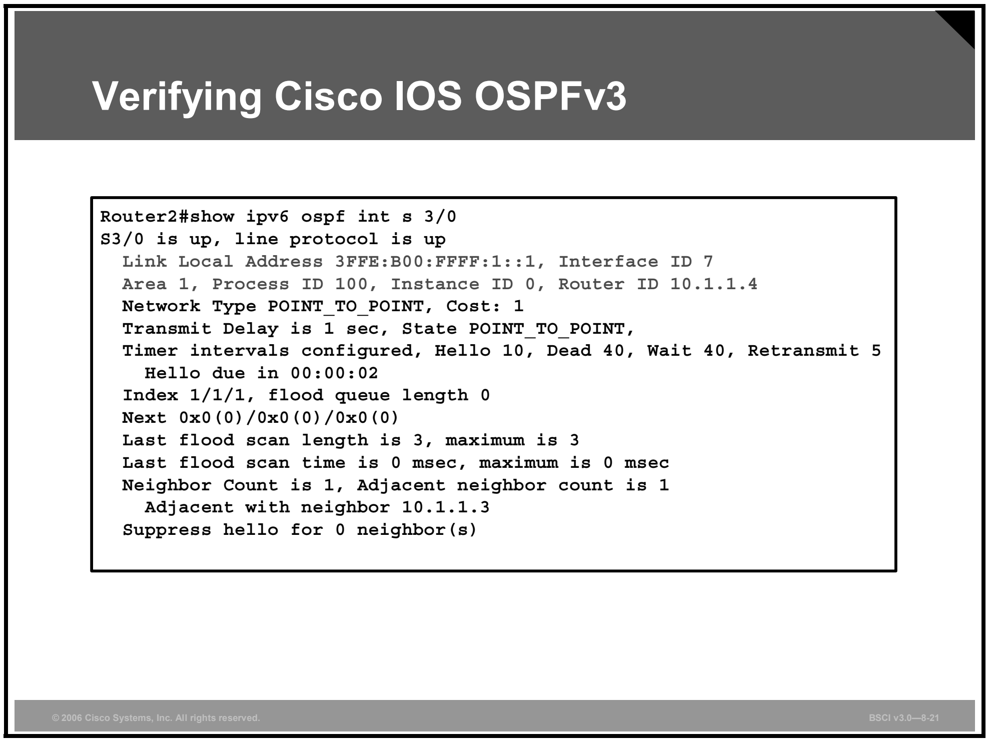
BSCI v3.0—8-20

The example shows an OSPF network of two routers, with an area 0 and area 1.

The interface-specific command **ipv6 ospf 100 area 0** will create the “ipv6 router ospf 100” process dynamically, as will the **ipv6 ospf 100 area 1** command.

Verifying OSPFv3

This topic explains how to verify OSPFv3.



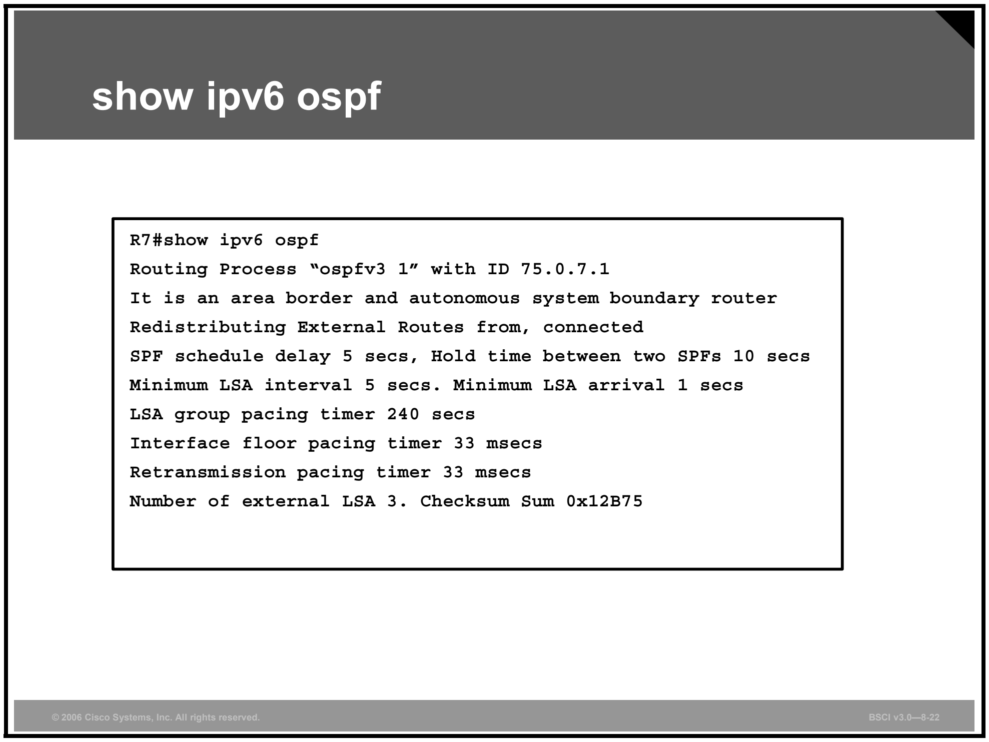
Verifying Cisco IOS OSPFv3

```
Router2#show ipv6 ospf int s 3/0
S3/0 is up, line protocol is up
  Link Local Address 3FFE:B00:FFFF:1::1, Interface ID 7
  Area 1, Process ID 100, Instance ID 0, Router ID 10.1.1.4
  Network Type POINT_TO_POINT, Cost: 1
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
  Index 1/1/1, flood queue length 0
  Next 0x0(0)/0x0(0)/0x0(0)
  Last flood scan length is 3, maximum is 3
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.1.1.3
  Suppress hello for 0 neighbor(s)
```

BSCI v3.0—8-21

There are several commonly used OSPFv3 **show** commands, including the **show ipv6 ospf** [*process-id*][*area-id*] **interface** [*interface*] command. This command displays OSPF-related interface information, as displayed in the figure.

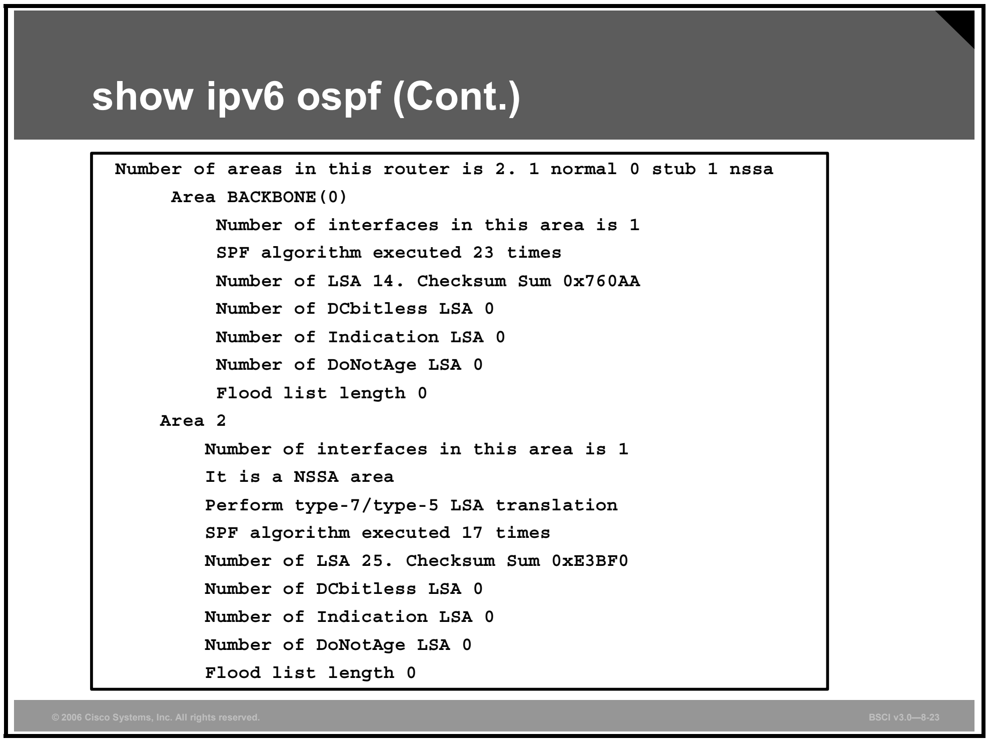
The **clear ipv6 ospf** [*process-id*] {**process** | **force-spf** | **redistribution** | **counters** [**neighbor** [*neighbor-interface* | *neighbor-id*]]} command triggers SPF recalculation and repopulation of the Routing Information Base (RIB).



show ipv6 ospf

```
R7#show ipv6 ospf
Routing Process “ospfv3 1” with ID 75.0.7.1
It is an area border and autonomous system boundary router
Redistributing External Routes from, connected
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
LSA group pacing timer 240 secs
Interface floor pacing timer 33 msecs
Retransmission pacing timer 33 msecs
Number of external LSA 3. Checksum Sum 0x12B75
```

BSCI v3.0—8-22



show ipv6 ospf (Cont.)

```
Number of areas in this router is 2. 1 normal 0 stub 1 nssa
     Area BACKBONE(0)
         Number of interfaces in this area is 1
         SPF algorithm executed 23 times
         Number of LSA 14. Checksum Sum 0x760AA
         Number of DCbitless LSA 0
         Number of Indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
    Area 2
        Number of interfaces in this area is 1
        It is a NSSA area
        Perform type-7/type-5 LSA translation
        SPF algorithm executed 17 times
        Number of LSA 25. Checksum Sum 0xE3BF0
        Number of DCbitless LSA 0
        Number of Indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0
```

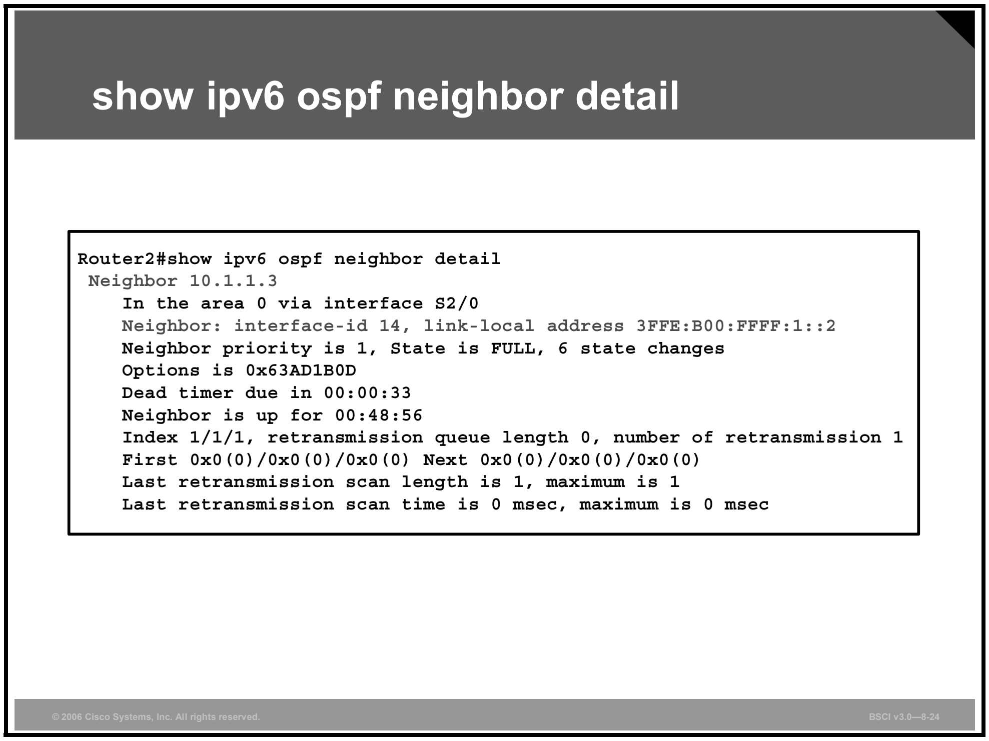
BSCI v3.0—8-23

The **show ipv6 ospf** [*process-id*][*area-id*] command displays general information about OSPF processes, as shown in the figures. The output in the second figure is a continuation of the output in the first figure.

This table provides some of the **show ipv6 ospf** command output fields and descriptions.

show ipv6 ospf Field Descriptions

Field	Description
Routing process "ospfv3 1" with ID 172.16.3.3	Process ID and OSPF router ID
LSA group pacing timer	Configured LSA group pacing timer (in seconds)
Interface flood pacing timer	Configured LSA flood pacing timer (in millisseconds)
Retransmission pacing timer	Configured LSA retransmission pacing timer (in milliseconds)
Number of areas	Number of areas in router, area addresses, and so on



show ipv6 ospf neighbor detail

```
Router2#show ipv6 ospf neighbor detail
 Neighbor 10.1.1.3
    In the area 0 via interface S2/0
    Neighbor: interface-id 14, link-local address 3FFE:B00:FFFF:1::2
    Neighbor priority is 1, State is FULL, 6 state changes
    Options is 0x63AD1B0D
    Dead timer due in 00:00:33
    Neighbor is up for 00:48:56
    Index 1/1/1, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

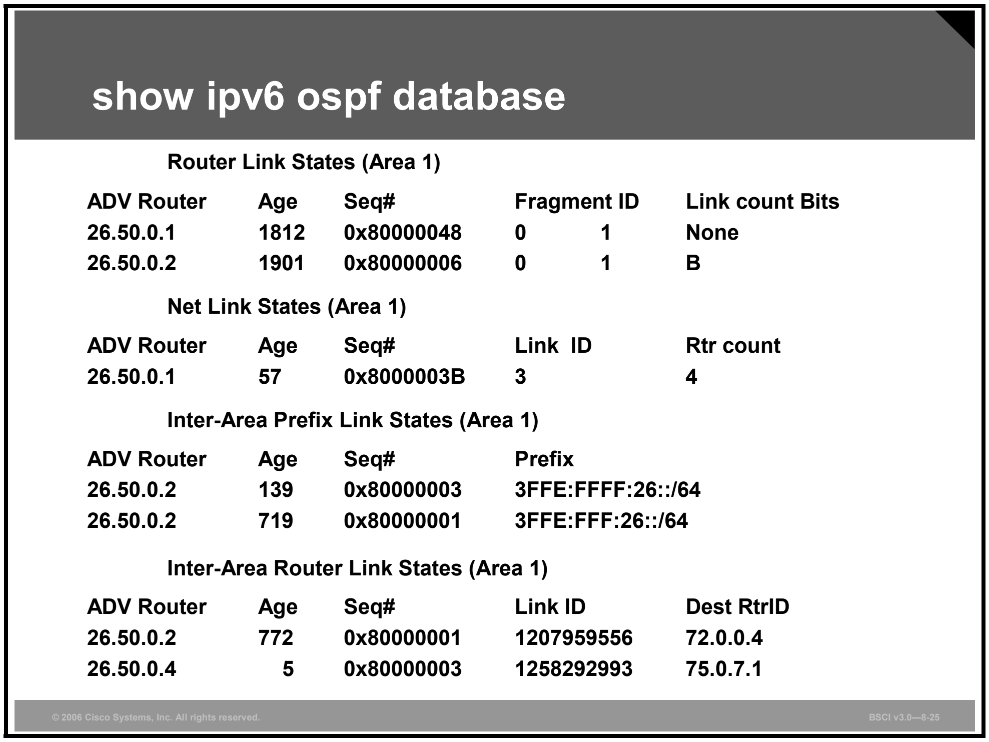
BSCI v3.0—8-24

The **show ipv6 ospf neighbor detail** command provides detailed information about IPv6 OSPF neighbors, as illustrated in the figure.

This table provides **show ipv6 ospf neighbor** command output fields and descriptions.

show ipv6 ospf neighbor detail Field Descriptions

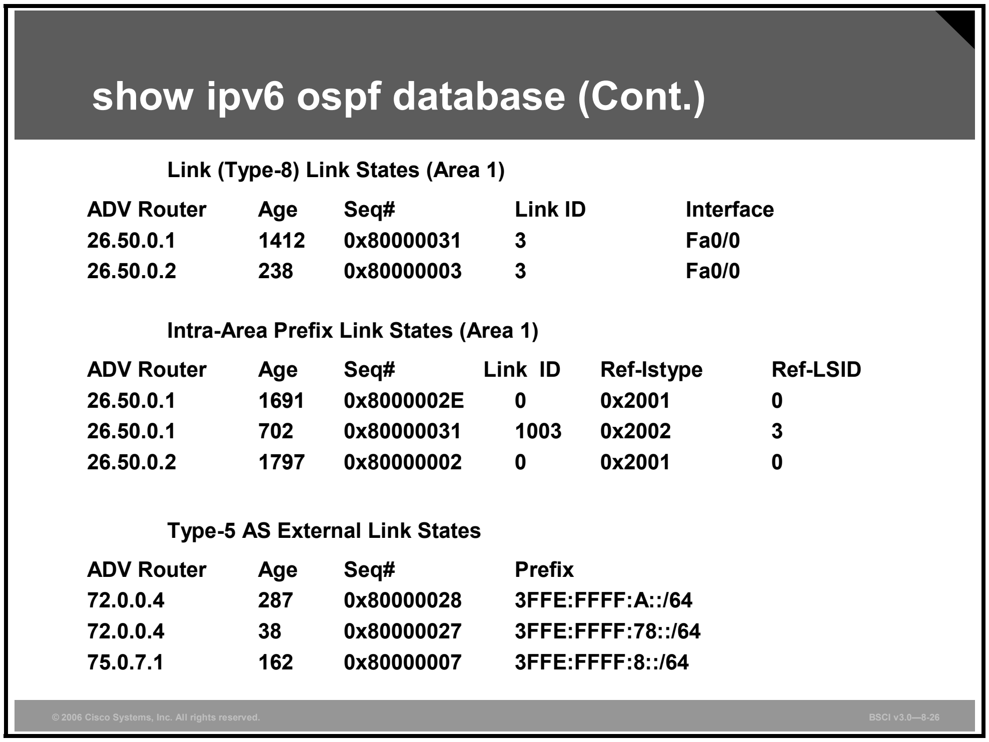
Field	Description
Neighbor ID; neighbor	Neighbor router ID.
In the area	Area and interface through which the OSPF neighbor is known.
Pri; Neighbor priority	Router priority of the neighbor, neighbor state.
State	OSPF state.
State changes	Number of state changes since the neighbor was created.
Options	Hello packet options field contents. (External bit [e-bit] only. Possible values are 0 and 2; 2 indicates that the area is not a stub, and 0 indicates that the area is a stub.)
Dead timer due in	Expected time before Cisco IOS software declares the neighbor dead.
Neighbor is up for	Number of hours, minutes, and seconds since the neighbor went into a two-way state (in HH:MM:SS format).
Index	Neighbor location in the area-wide and AS-wide retransmission queue.
Retransmission queue length	Number of elements in the retransmission queue.
Number of retransmission	Number of times that update packets have been resent during flooding.
First	Memory location of the flooding details.
Next	Memory location of the flooding details.
Last retransmission scan length	Number of LSAs in the last retransmission packet.
Maximum	Maximum number of LSAs sent in any retransmission packet.
Last retransmission scan time	Time taken to build the last retransmission packet.
Maximum	Maximum time taken to build any retransmission packet.



show ipv6 ospf database

```
            Router Link States (Area 1)

ADV Router     Age     Seq#           Fragment ID     Link count Bits
26.50.0.1      1812    0x80000048     0        1      None
26.50.0.2      1901    0x80000006     0        1      B

            Net Link States (Area 1)

ADV Router     Age     Seq#           Link  ID        Rtr count
26.50.0.1      57      0x8000003B     3               4

            Inter-Area Prefix Link States (Area 1)

ADV Router     Age     Seq#           Prefix
26.50.0.2      139     0x80000003     3FFE:FFFF:26::/64
26.50.0.2      719     0x80000001     3FFE:FFF:26::/64

            Inter-Area Router Link States (Area 1)

ADV Router     Age     Seq#           Link ID         Dest RtrID
26.50.0.2      772     0x80000001     1207959556      72.0.0.4
26.50.0.4        5     0x80000003     1258292993      75.0.7.1
```

© 2006 Cisco Systems, Inc. All rights reserved.   BSCI v3.0—8-25



show ipv6 ospf database (Cont.)

```
            Link (Type-8) Link States (Area 1)

ADV Router     Age     Seq#           Link ID         Interface
26.50.0.1      1412    0x80000031     3               Fa0/0
26.50.0.2      238     0x80000003     3               Fa0/0

            Intra-Area Prefix Link States (Area 1)

ADV Router     Age     Seq#         Link  ID   Ref-lstype      Ref-LSID
26.50.0.1      1691    0x8000002E     0        0x2001          0
26.50.0.1      702     0x80000031     1003     0x2002          3
26.50.0.2      1797    0x80000002     0        0x2001          0

            Type-5 AS External Link States

ADV Router     Age     Seq#           Prefix
72.0.0.4       287     0x80000028     3FFE:FFFF:A::/64
72.0.0.4       38      0x80000027     3FFE:FFFF:78::/64
75.0.7.1       162     0x80000007     3FFE:FFFF:8::/64
```

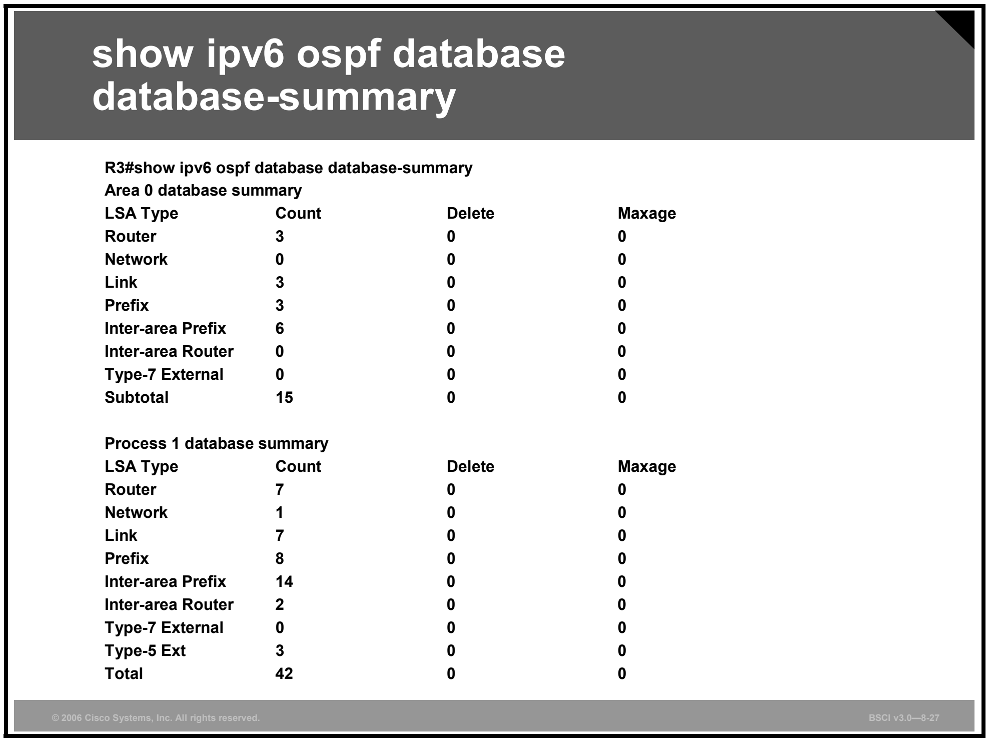
© 2006 Cisco Systems, Inc. All rights reserved.   BSCI v3.0—8-26

The two figures illustrate sample partial output from the **show ipv6 ospf database** command. The output in the second figure is a continuation of the output in the first figure.

This table provides **show ipv6 ospf database** command output field descriptions.

show ipv6 ospf database Field Descriptions

Field	Description
ADV Router	Advertising router ID
Age	Link-state age
Seq#	Link-state sequence number (detects old or duplicate LSAs)
Link ID	Interface ID number
Ref-lstype	Referenced link-state type



show ipv6 ospf database database-summary

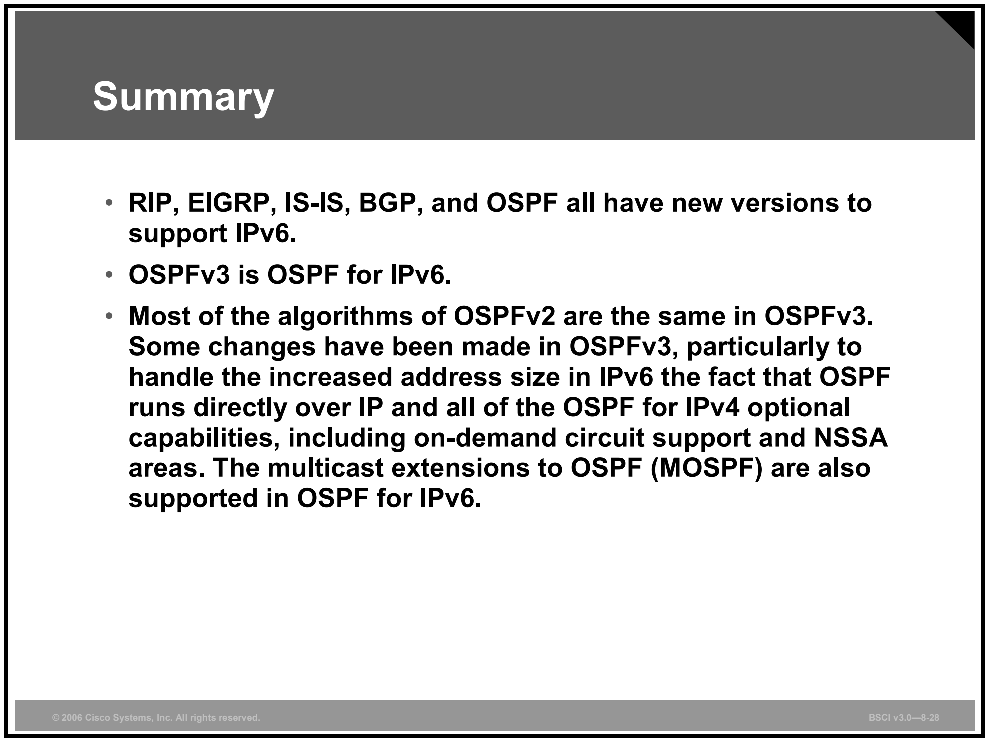
```
R3#show ipv6 ospf database database-summary
Area 0 database summary
LSA Type          Count          Delete          Maxage
Router            3              0               0
Network           0              0               0
Link              3              0               0
Prefix            3              0               0
Inter-area Prefix 6              0               0
Inter-area Router 0              0               0
Type-7 External   0              0               0
Subtotal          15             0               0

Process 1 database summary
LSA Type          Count          Delete          Maxage
Router            7              0               0
Network           1              0               0
Link              7              0               0
Prefix            8              0               0
Inter-area Prefix 14             0               0
Inter-area Router 2              0               0
Type-7 External   0              0               0
Type-5 Ext        3              0               0
Total             42             0               0
```

BSCI v3.0—8-27

This figure illustrates sample output from the **show ipv6 ospf database database-summary** command**.**

Summary

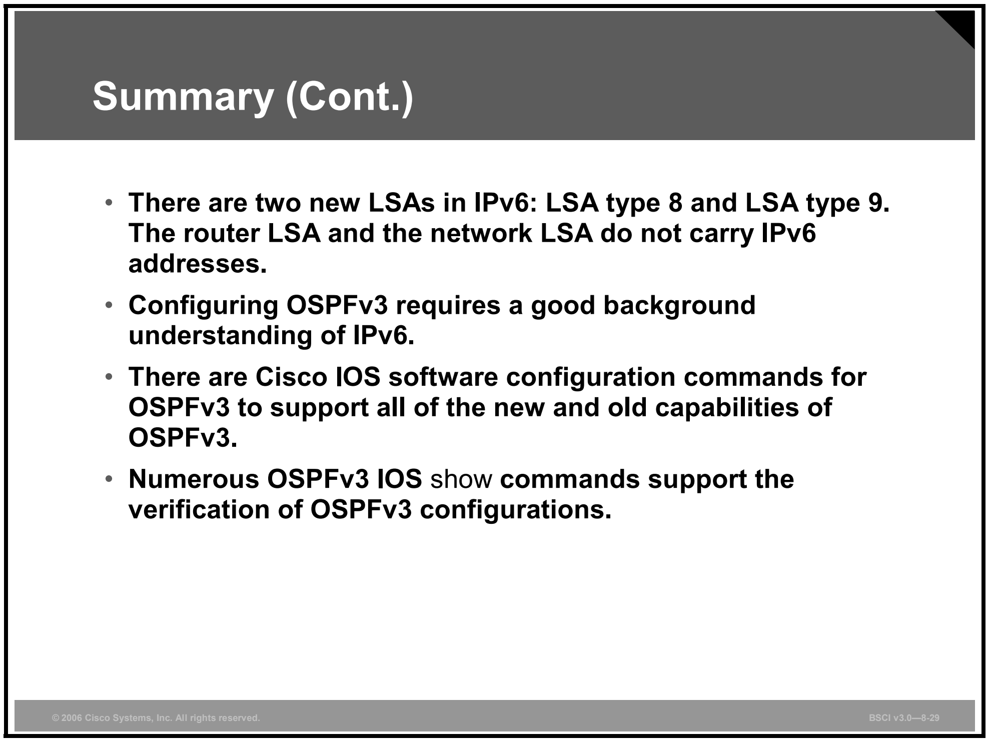
This topic summarizes the key points that were discussed in this lesson.



Summary

- RIP, EIGRP, IS-IS, BGP, and OSPF all have new versions to support IPv6.
- OSPFv3 is OSPF for IPv6.
- Most of the algorithms of OSPFv2 are the same in OSPFv3. Some changes have been made in OSPFv3, particularly to handle the increased address size in IPv6 the fact that OSPF runs directly over IP and all of the OSPF for IPv4 optional capabilities, including on-demand circuit support and NSSA areas. The multicast extensions to OSPF (MOSPF) are also supported in OSPF for IPv6.

BSCI v3.0—8-28



Summary (Cont.)

- There are two new LSAs in IPv6: LSA type 8 and LSA type 9. The router LSA and the network LSA do not carry IPv6 addresses.
- Configuring OSPFv3 requires a good background understanding of IPv6.
- There are Cisco IOS software configuration commands for OSPFv3 to support all of the new and old capabilities of OSPFv3.
- Numerous OSPFv3 IOS show commands support the verification of OSPFv3 configurations.

BSCI v3.0—8-29

Lesson 5

Using IPv6 with IPv4

Overview

Now that you have seen the process for enabling IP version 6 (IPv6), you need to identify methods allowing integration and coexistence of IP version 4 (IPv4) and IPv6 on networks.

The successful market adoption of any new technology depends on its easy integration with the existing infrastructure without significant disruption of services. The Internet consists of hundreds of thousands of IPv4 networks and millions of IPv4 nodes. The challenge lies in making the integration and transition as transparent as possible to the end users.

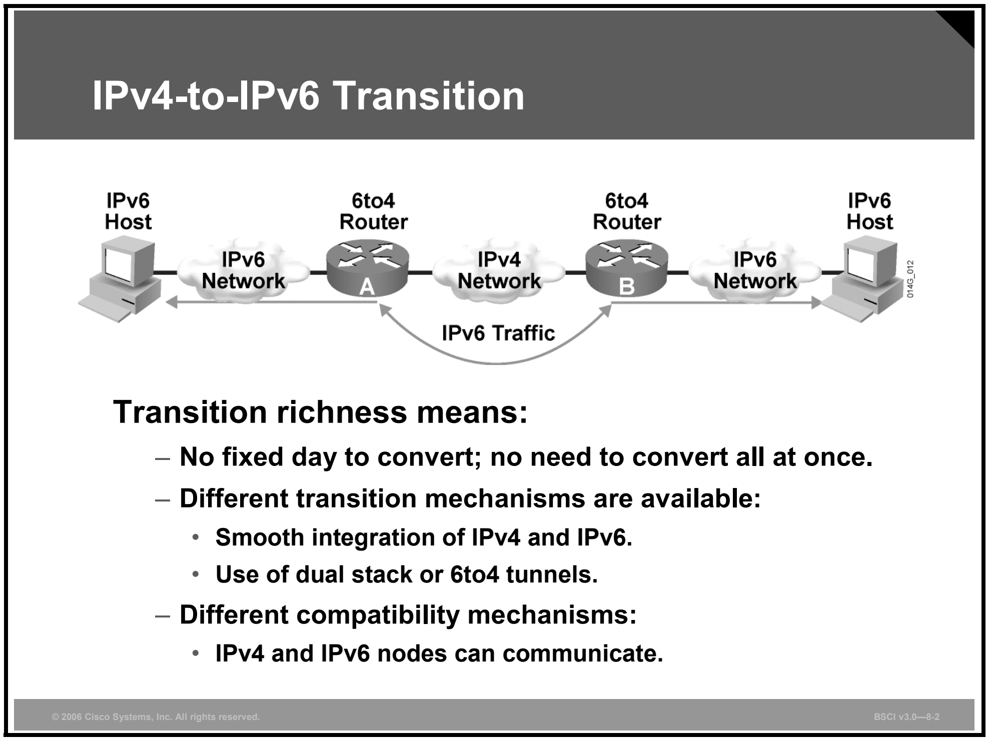
Objectives

Upon completing this topic, you will be able to identify IPv6 integration and coexistence methods. This ability includes being able to meet these objectives:

- Describe each of the transition mechanisms used by IPv6 network traffic to transit IPv4 networks
- Describe how IPv6-over-IPv4 encapsulation (tunneling) works and explain how to express IPv4 addresses in IPv6 format

Describing IPv6-to-IPv4 Transition Mechanisms

This topic describes the transition mechanisms used by IPv6 network traffic to transit IPv4 networks.



IPv4-to-IPv6 Transition

Transition richness means:

- **No fixed day to convert; no need to convert all at once.**
- **Different transition mechanisms are available:**
  - **Smooth integration of IPv4 and IPv6.**
  - **Use of dual stack or 6to4 tunnels.**
- **Different compatibility mechanisms:**
  - **IPv4 and IPv6 nodes can communicate.**

BSCI v3.0—8-2

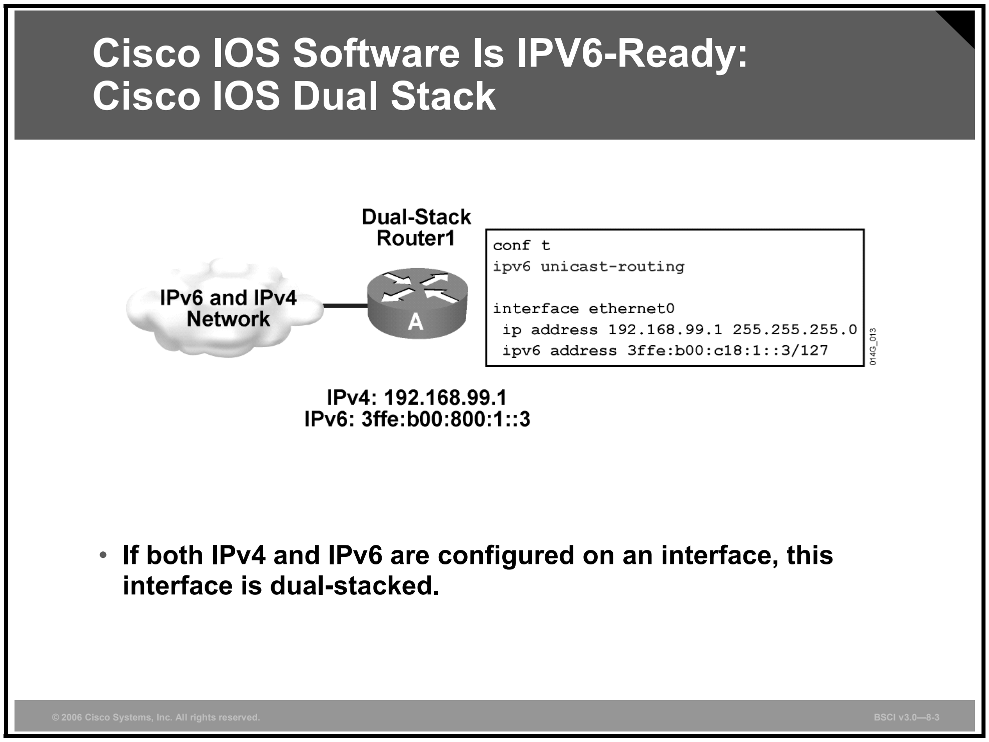
The transition from IPv4 does not require upgrades on all nodes at the same time. Many transition mechanisms enable smooth integration of IPv4 and IPv6. Other mechanisms that allow IPv4 nodes to communicate with IPv6 nodes are available. All of these mechanisms are applied to different situations.

The two most common techniques to transition from IPv4 to IPv6 are as follows:

- Dual stack
- IPv6-over-IPv4 (6to4) tunnels

For communication between IPv4 and IPv6 networks, there can be IPv4 addresses that are encapsulated in an IPv6 address.

The figure shows an example of a transition and integration mechanism. The 6to4 routers automatically encapsulate the IPv6 traffic inside IPv4 packets.



Cisco IOS Software Is IPV6-Ready: Cisco IOS Dual Stack

```
conf t
ipv6 unicast-routing

interface ethernet0
 ip address 192.168.99.1 255.255.255.0
 ipv6 address 3ffe:b00:c18:1::3/127
```

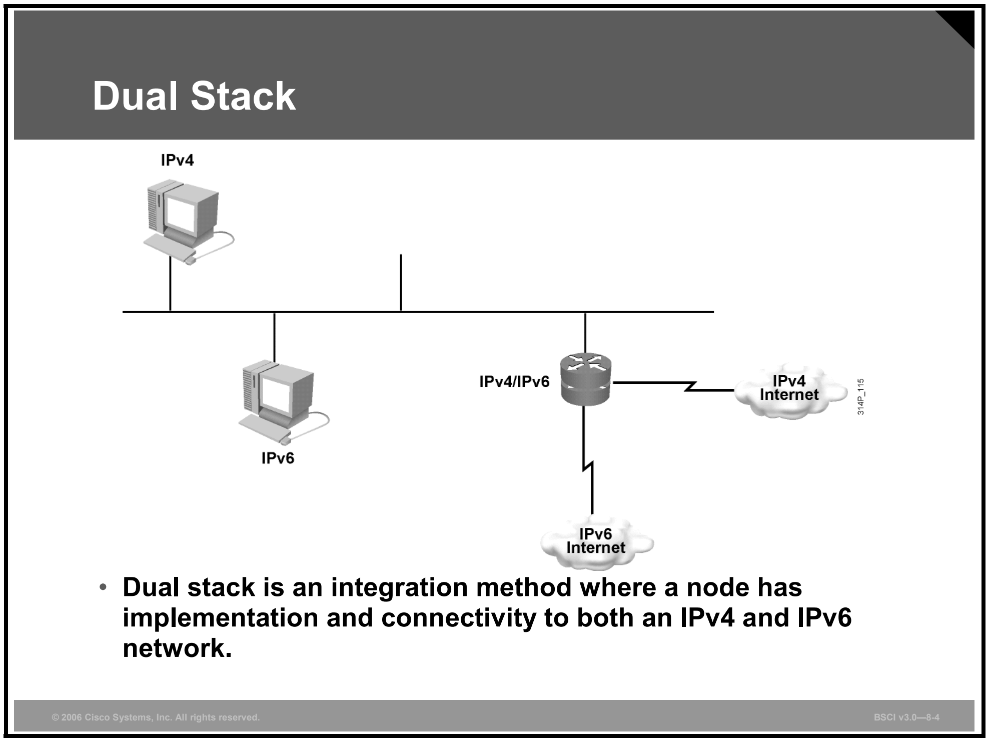
- **If both IPv4 and IPv6 are configured on an interface, this interface is dual-stacked.**

Cisco IOS software is IPv6-ready. As soon as IPv4 and IPv6 basic configurations are complete on the interface, the interface is dual-stacked, and it forwards IPv4 and IPv6 traffic.

Using IPv6 on a Cisco IOS router requires that you use the global configuration command **ipv6 unicast-routing**. This command enables the forwarding of IPv6 datagrams. All interfaces that forward IPv6 traffic must have an IPv6 address. The interface command is as follows:

ipv6 address *IPv6-address* [*/prefix length*]

This command specifies an IPv6 network assigned to the interface and enables IPv6 processing on the interface.

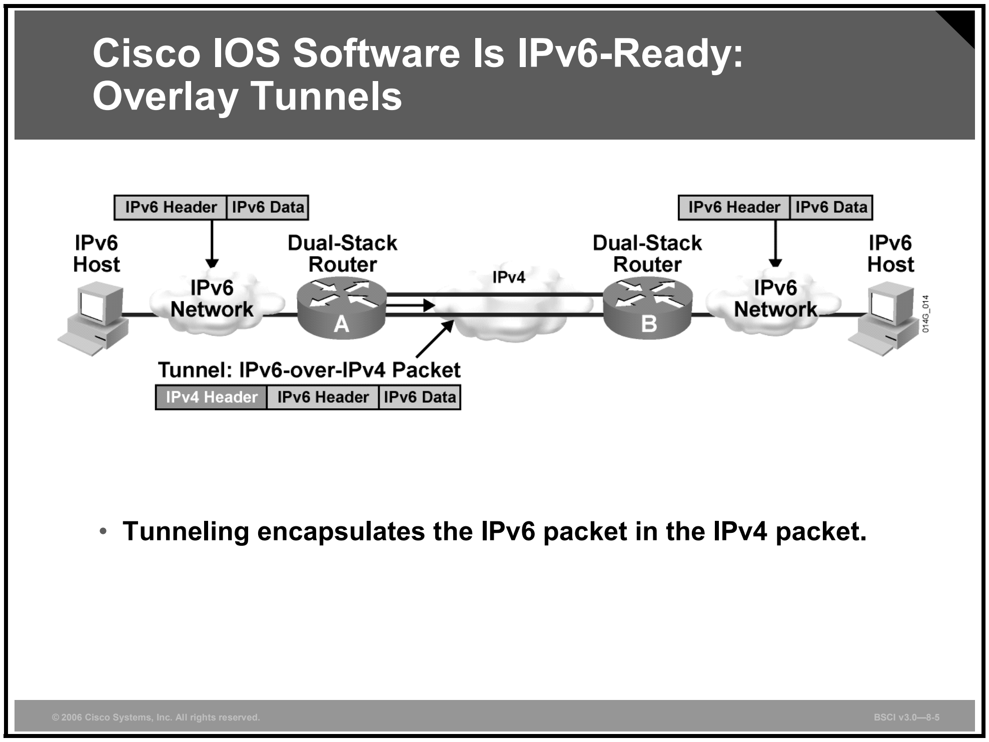


Dual Stack

- **Dual stack is an integration method where a node has implementation and connectivity to both an IPv4 and IPv6 network.**

Dual stack is an integration method where a node has implementation and connectivity to both an IPv4 and IPv6 network, and thus the node has two stacks. This configuration can be accomplished on the same interface or on multiple interfaces.

- A dual-stack node chooses which stack to use based on destination address. A dual-stack node should prefer IPv6 when available. The dual-stack approach to IPv6 integration, in which nodes have both IPv4 and IPv6 stacks, will be one of the most commonly used integration methods. Old IPv4-only applications continue to work as before. New and modified applications take advantage of both IP layers.
- A new application programming interface (API) is defined to support both IPv4 and IPv6 addresses and Domain Name System (DNS) requests. This new API replaces the gethostbyname and gethostbyaddr calls. A converted application will be able to make use of both IPv4 and IPv6. An application can be converted to the new API while still using only IPv4.
- Past experience in porting IPv4 applications to IPv6 suggests that, for most applications, it is a minimal change in some localized places inside the source code. This technique is well known and has been applied in the past for other protocol transitions. It enables gradual application upgrades, one by one, to IPv6.



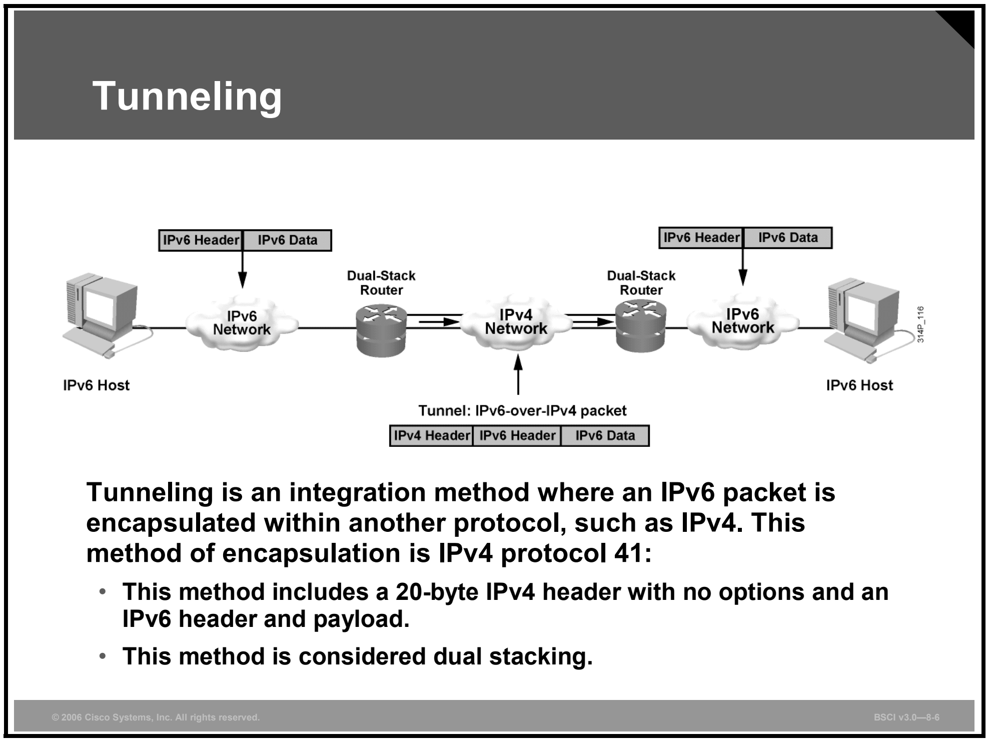
Cisco IOS Software Is IPv6-Ready: Overlay Tunnels

- **Tunneling encapsulates the IPv6 packet in the IPv4 packet.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-5

Networking often uses tunnels to overlay an incompatible functionality on an existing network. Tunneling IPv6 traffic over an IPv4 network requires one edge router to encapsulate the IPv6 packet inside an IPv4 packet and another router to decapsulate it.

This process enables you to connect the IPv6 islands without converting the entire network to IPv6.



Tunneling

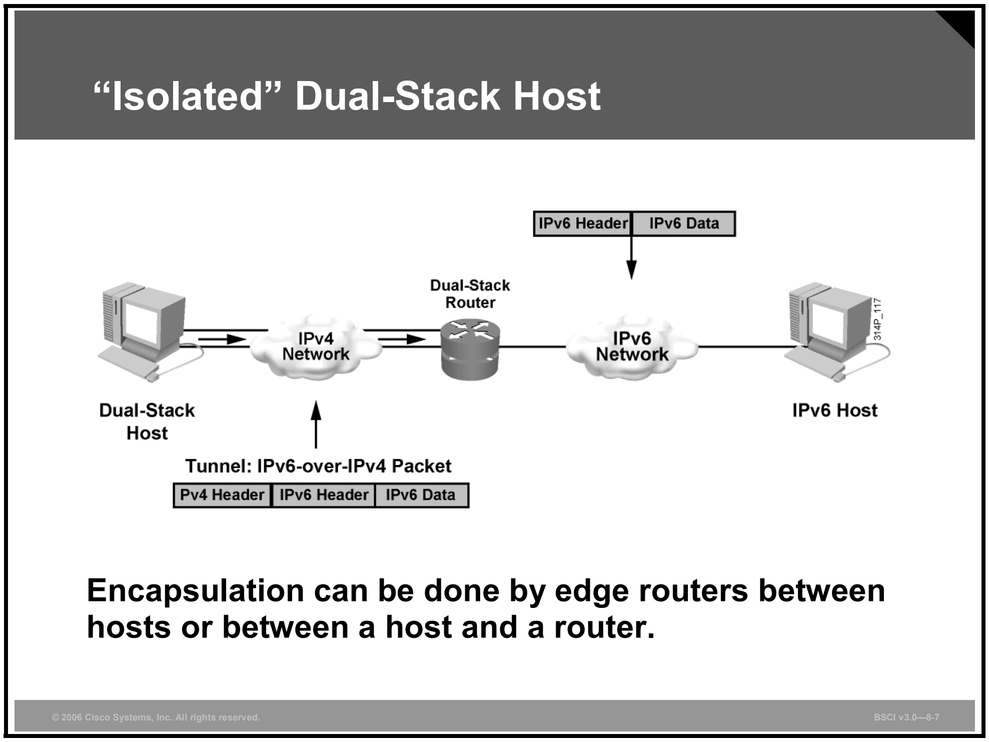
Tunneling is an integration method where an IPv6 packet is encapsulated within another protocol, such as IPv4. This method of encapsulation is IPv4 protocol 41:

- **This method includes a 20-byte IPv4 header with no options and an IPv6 header and payload.**
- **This method is considered dual stacking.**

BSCI v3.0—8-6

Tunneling is an integration method where an IPv6 packet is encapsulated within another protocol, such as IPv4. This method of encapsulation is IPv4 protocol 41:

- This method includes a 20-byte IPv4 header with no options and an IPv6 header and payload.
- This method is considered dual stacking. This process enables the connection of IPv6 islands without the need to also convert an intermediary network to IPv6. Tunneling presents these two issues:
  - The maximum transmission unit (MTU) is effectively decreased by 20 octets (if the IPv4 header does not contain any optional field).
  - A tunneled network is often difficult to troubleshoot. Tunneling is an intermediate integration and transition technique that should not be considered a final solution. Native IPv6 architecture should be the ultimate goal.



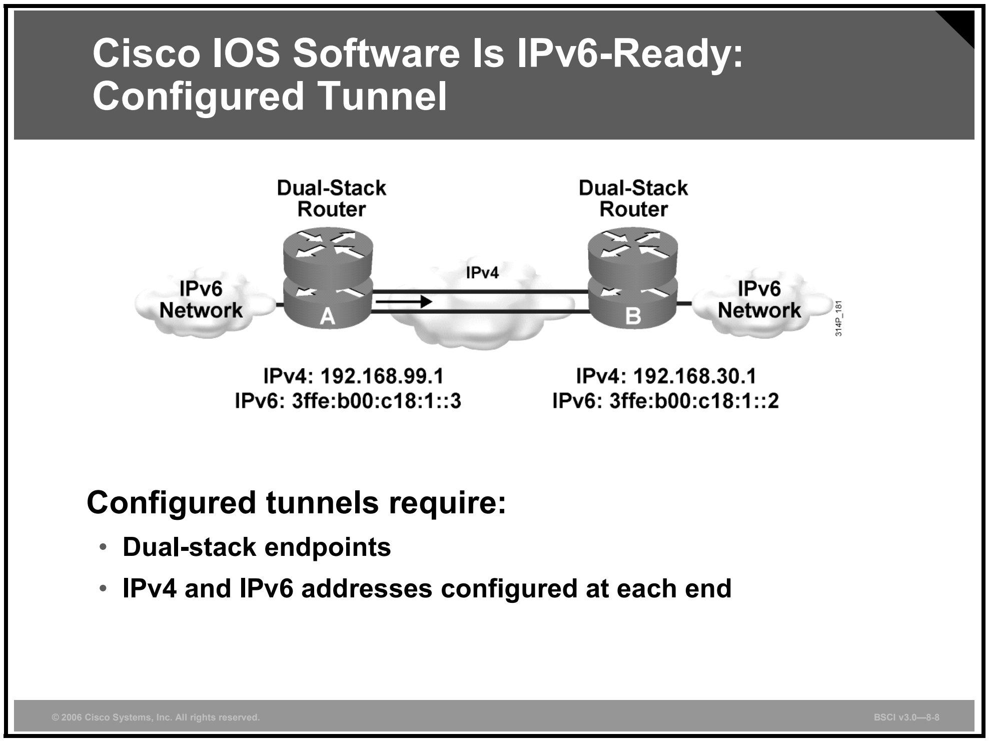
“Isolated” Dual-Stack Host

Encapsulation can be done by edge routers between hosts or between a host and a router.

BSCI v3.0—8-7

Encapsulation can be done by edge routers between hosts or between a host and a router. This example shows an isolated dual-stack host using an encapsulated tunnel to connect to the edge router of the IPv6 network.

Tunneling will not work if an intermediary node between the two end points of the tunnel, such as a firewall, filters out IPv4 protocol 41, which is the IPv6-over-IPv4 encapsulation.



Cisco IOS Software Is IPv6-Ready: Configured Tunnel

Configured tunnels require:

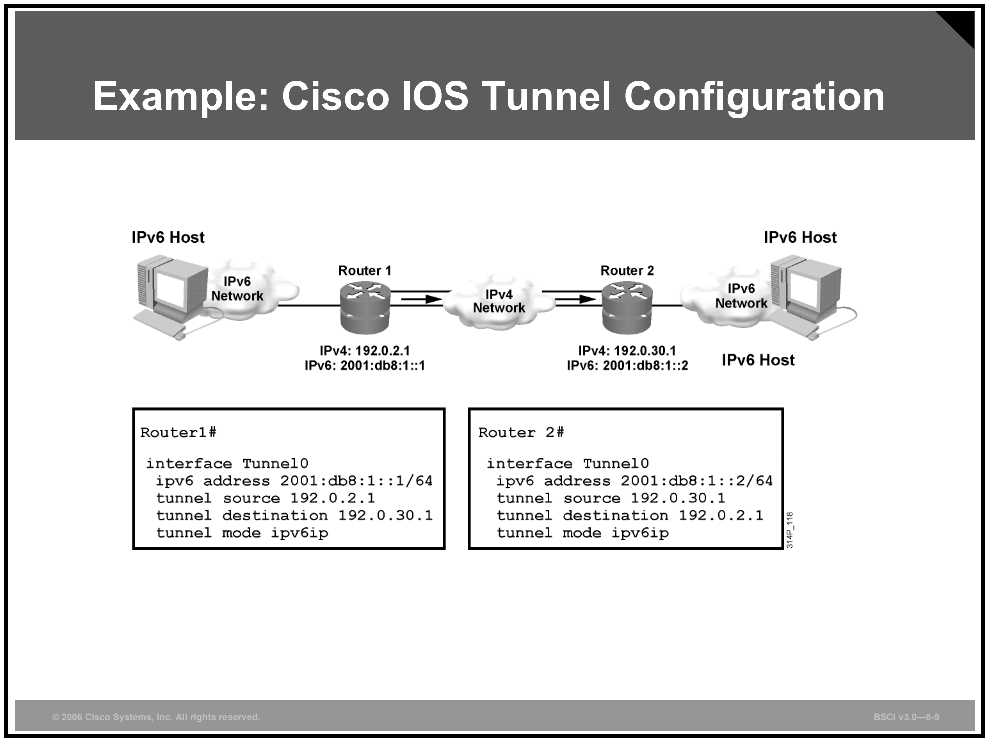
- **Dual-stack endpoints**
- **IPv4 and IPv6 addresses configured at each end**

BSCI v3.0—8-8

In a manually configured tunnel, you configure both the IPv4 and IPv6 addresses statically. Perform this configuration on the routers at each end of the tunnel.

These end routers must be dual stacked, and the configuration cannot change dynamically as network and routing needs change. Routing must be set up properly to forward a packet between the two IPv6 networks.

Tunnel endpoints can be unnumbered, but unnumbered endpoints make troubleshooting difficult. The IPv4 practice of saving addresses for tunnel endpoints is no longer an issue.



Example: Cisco IOS Tunnel Configuration

```
Router1#

 interface Tunnel0
  ipv6 address 2001:db8:1::1/64
  tunnel source 192.0.2.1
  tunnel destination 192.0.30.1
  tunnel mode ipv6ip
```

```
Router 2#

 interface Tunnel0
  ipv6 address 2001:db8:1::2/64
  tunnel source 192.0.30.1
  tunnel destination 192.0.2.1
  tunnel mode ipv6ip
```

The example shows the configuration of two Cisco Systems routers connected with IPv6-over-IPv4 encapsulation. The command **interface Tunnel0** is used, with the tunnel source and destination specified with the underlying network addresses, which are IPv4 addresses. A static IPv6 address is configured on the Tunnel0 interface. The command that enables the 6to4 tunneling is **tunnel mode ipv6ip**.

Other Tunneling and Transition Mechanisms

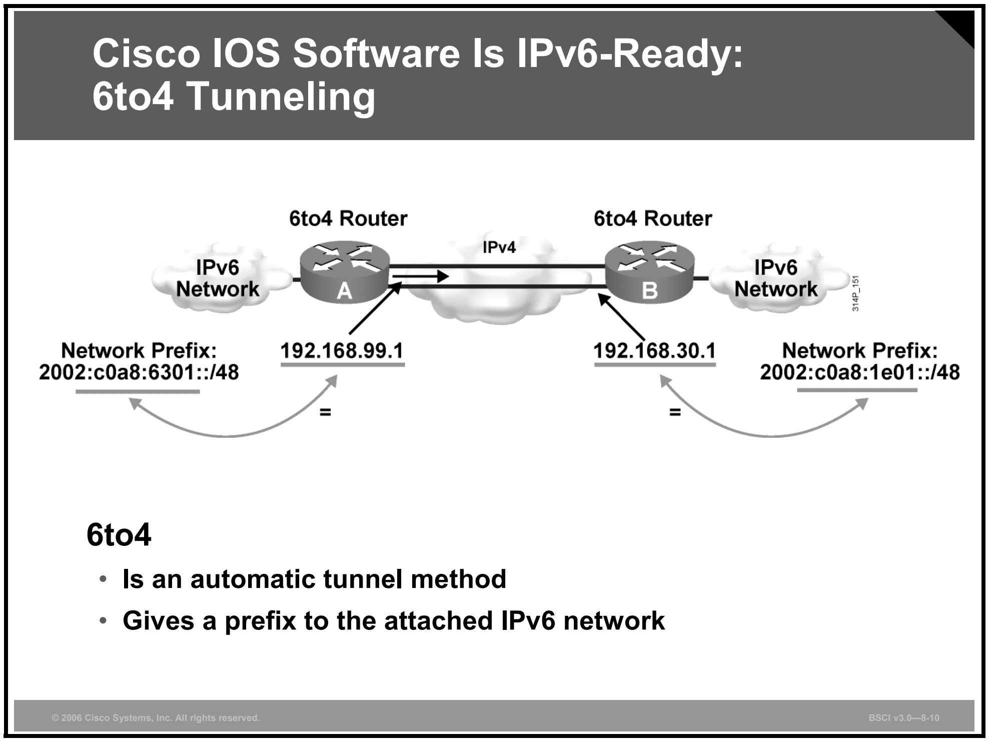
Several other automatic tunneling transition mechanisms exist, including these:

- **6to4:** This mechanism uses the reserved prefix 2002::/16 to allow an IPv4 Internet-connected site to create and use a /48 IPv6 prefix based on a single globally routable or reachable IPv4 address.
- **Intra-Site Automatic Tunnel Addressing Protocol (ISATAP):** ISATAP allows an IPv4 private intranet (which may or may not be using RFC 1918 addresses) to incrementally implement IPv6 nodes without upgrading the network.

Another transition mechanism is Teredo (formerly known as Shipworm). This mechanism tunnels IPv6 datagrams within IPv4 User Datagram Protocol (UDP). This method provides for private IPv4 address use and IPv4 Network Address Translation (NAT) traversal.

Describing IPv6-over-IPv4 Tunneling Mechanisms and IPv4 Addresses in IPv6 Format

This topic describes how IPv6-over-IPv4 encapsulation works and how to express IPv4 addresses in IPv6 format.



Cisco IOS Software Is IPv6-Ready: 6to4 Tunneling

6to4

- Is an automatic tunnel method
- Gives a prefix to the attached IPv6 network

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-10

The 6to4 tunneling method automatically establishes the connection of IPv6 islands through an IPv4 network. The 6to4 tunneling method applies a valid IPv6 prefix to each IPv6 island, which enables the fast deployment of IPv6 in a corporate network without address retrieval from the Internet service providers (ISPs) or registries.

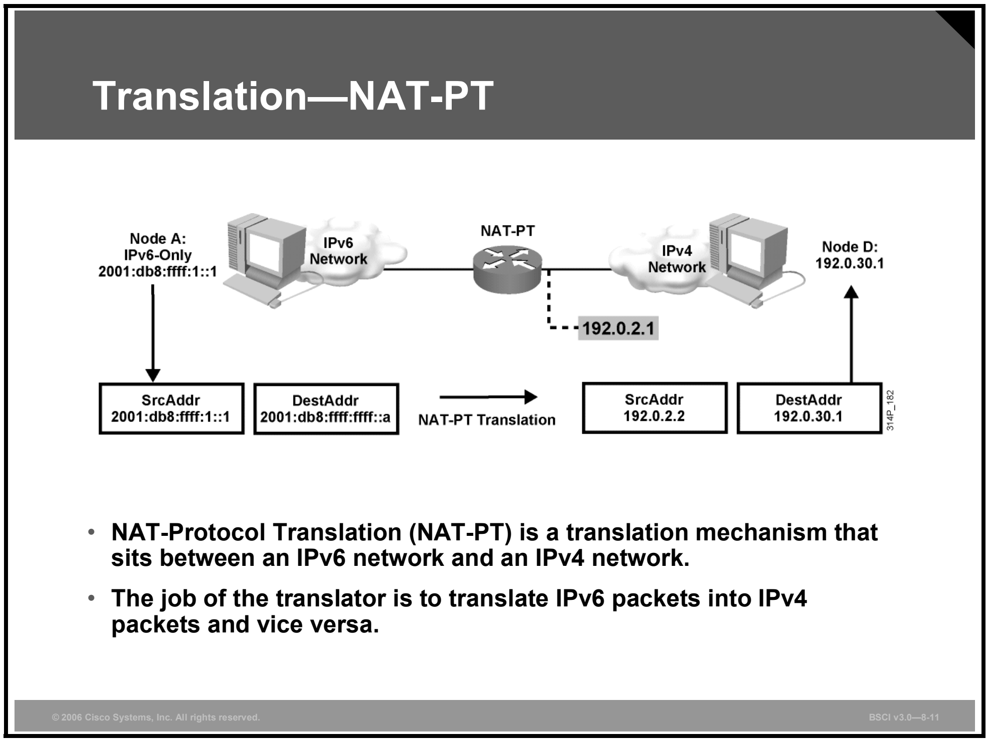
The 6to4 tunneling method requires a special code on the edge routers, but the IPv6 hosts and routers inside the 6to4 site do not require new features to support 6to4. Each 6to4 site receives a /48 prefix, which is the concatenation of 0x2002 and the hexadecimal IPv4 address of the edge router.

For example, if the IPv4 address of the edge router is 192.168.99.1, the prefix of its IPv6 network is 2002:c0a8:6301::/48, because c0a86301 is the hexadecimal representation of 192.168.99.1. The IPv6 network can substitute any IP address in the space after the first 16-bit section (0x2002).

When an IPv6 packet with a destination address in the range of 2002::/16 reaches the 6to4 edge router, the 6to4 edge router extracts the IPv4 address that is embedded in the 2002:: destination address (inserted between the third and sixth octets, inclusive). The 6to4 router then encapsulates the IPv6 packet in an IPv4 packet with the destination IPv4 address that was extracted from inside the IPv6 destination address.

This IPv4 address represents the address of the other 6to4 edge router of the destination 6to4 site. The destination edge router decapsulates the IPv6 packet in the IPv4 packet and then forwards the native packet toward its final destination.

Note	2002::/16 is the address range specifically assigned to 6to4.



Translation—NAT-PT

Node A:
IPv6-Only
2001:db8:ffff:1::1

IPv6 Network

NAT-PT

IPv4 Network

Node D:
192.0.30.1

192.0.2.1

SrcAddr 2001:db8:ffff:1::1

DestAddr 2001:db8:ffff:ffff::a

NAT-PT Translation

SrcAddr 192.0.2.2

DestAddr 192.0.30.1

314P_182

- NAT-Protocol Translation (NAT-PT) is a translation mechanism that sits between an IPv6 network and an IPv4 network.
- The job of the translator is to translate IPv6 packets into IPv4 packets and vice versa.

For legacy equipment that will not be upgraded to IPv6 and for some deployment scenarios, techniques that can connect IPv4-only nodes on IPv6-only nodes are available. Translation is basically an extension of NAT techniques.

NAT-PT

NAT-Protocol Translation (NAT-PT) is a translation mechanism that sits between an IPv6 network and an IPv4 network. The job of the translator is to translate IPv6 packets into IPv4 packets and vice versa.

The Stateless IP/Internet Control Message Protocol (ICMP) Translation (SIIT) algorithm translates the IP header fields. NAT handles the IP address translation. This example shows static NAT-PT translations. NAT-PT translations may also be mapped dynamically based on DNS queries using a DNS application level gateway (DNS ALG).

The example shows the translation of an IPv6 datagram sent from node A to node D. From the perspective of node A, it is establishing a communication to another IPv6 node. One advantage of NAT-PT is that no modifications are required on IPv6 node A; all it needs to know is the IPv6 address mapping to the IPv4 address of node D. This mapping can be obtained dynamically from the DNS server.

IPv4 node D can also send a datagram to node A by using the IPv4 address mapped to the IPv6 address of node A. Again, from the perspective of node D, it is establishing IPv4 communication with its correspondent. Again, node D requires no modification.

Other possible solutions are as follows:

- **ALGs:** This method uses a dual-stack approach and enables a host in an IPv6-only domain to send data to another host in an IPv4-only domain. It requires that all application servers on a gateway run IPv6.
- **API:** You can install a specific module in a host TCP/IP stack for every host on the network. The module intercepts IP traffic through an API and converts it for the IPv6 counterpart.

BIA and BIS

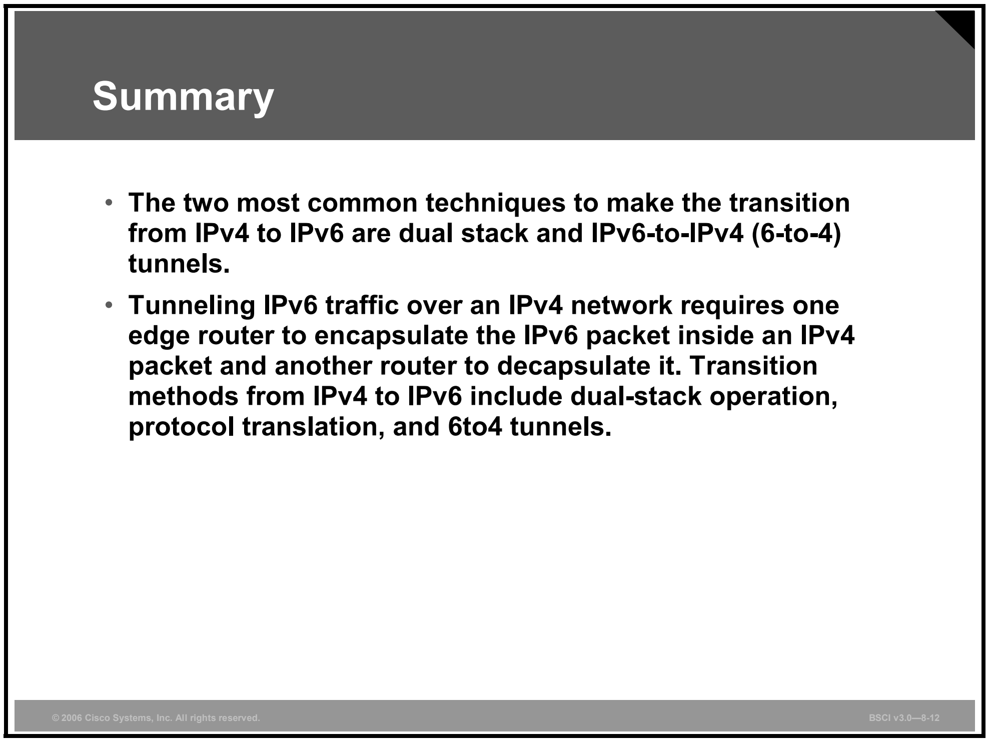
Bump-in-the-API (BIA) and Bump-in-the-Stack (BIS) are localized implementations of NAT-PT. They provide support for translation from upper layers that are IPv4-only down through the Open Systems Interconnection (OSI) layers to the stack.

These implementations intercept either API calls or packets in the stack and translate them on the fly. Only IPv6 packets will travel out on the wire. Also note that not all applications will work with BIA or BIS solutions.

FTP, which embeds IP addresses in the packet payload, would not work. The outer IP addresses and packets would be translated by BIA or BIS, but the embedded (IPv6) addresses would not (going back up the stack), which would not work for an IPv4-only FTP application.

Summary

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

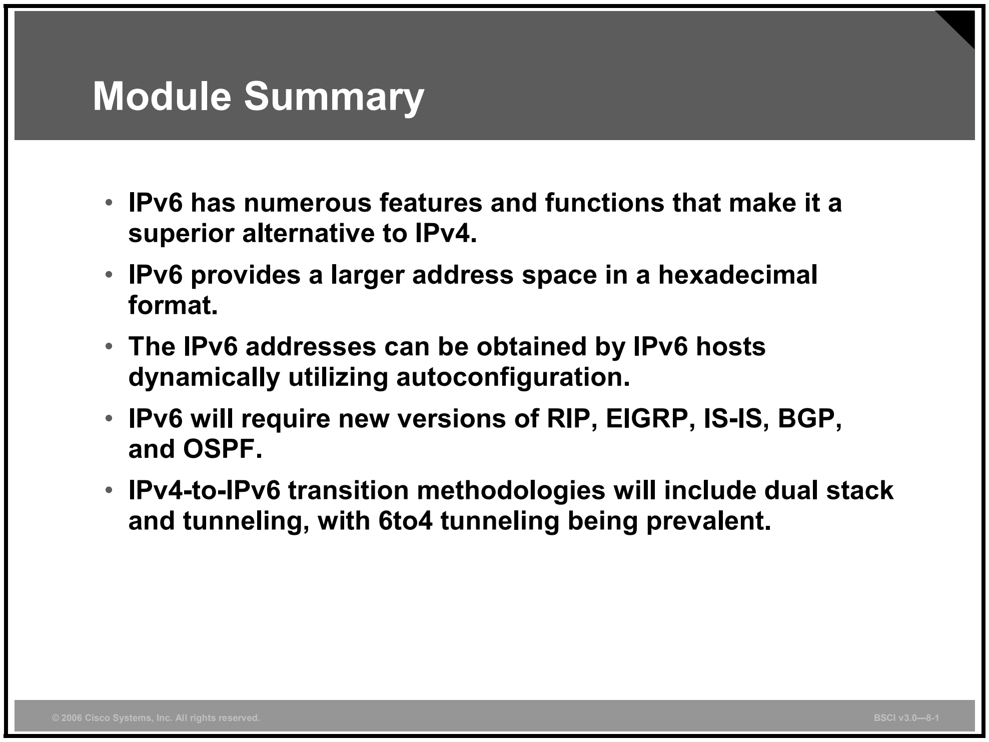


Summary

- The two most common techniques to make the transition from IPv4 to IPv6 are dual stack and IPv6-to-IPv4 (6-to-4) tunnels.
- Tunneling IPv6 traffic over an IPv4 network requires one edge router to encapsulate the IPv6 packet inside an IPv4 packet and another router to decapsulate it. Transition methods from IPv4 to IPv6 include dual-stack operation, protocol translation, and 6to4 tunnels.

Module Summary

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



Module Summary

- **IPv6 has numerous features and functions that make it a superior alternative to IPv4.**
- **IPv6 provides a larger address space in a hexadecimal format.**
- **The IPv6 addresses can be obtained by IPv6 hosts dynamically utilizing autoconfiguration.**
- **IPv6 will require new versions of RIP, EIGRP, IS-IS, BGP, and OSPF.**
- **IPv4-to-IPv6 transition methodologies will include dual stack and tunneling, with 6to4 tunneling being prevalent.**

© 2006 Cisco Systems, Inc. All rights reserved. BSCI v3.0—8-1

This module is an intense overview of IP version 6 (IPv6), beginning with why it will become the protocol of choice in the future and the benefits of that choice. The changes in the addressing format and the packet header format were discussed in detail, including autoconfiguration and the role of the multicast address.

A major portion of the module was devoted to describing routing IPv6. All possible routing protocols were defined and Open Shortest Path First Protocol (OSPF) for IPv6 was covered in more detail. The transition strategies to migrate from IPv4 to IPv6 were defined as well.

Throughout the module, Cisco IOS configuration, verification, and troubleshooting commands were shown.

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 of the following is NOT an advantage of IPv6 compared to IPv4? (Source: Introducing IPv6)

A) larger address space
B) shorter header
C) simpler header
D) support for IPsec on every link

Q2) Why is NAT not a requirement for IPv6? (Source: Introducing IPv6)

A) NAT is not available with IPv6.
B) IPv6 addresses do not have a private address space.
C) IPv6 allows all users in an enterprise to have a global address.
D) Hexadecimal addresses cannot be translated.

Q3) How will IPv6 enable smaller routing tables in Internet routers? (Source: Introducing IPv6)

A) defined aggregation points in the address space
B) a new routing protocol
C) autoconfiguration
D) site local addresses

Q4) How can consecutive chunks of zeros be condensed in an IPv6 address? (Source: Defining IPv6 Addressing)

A) with the ":::" symbol
B) by eliminating leading zeros
C) by replacing each four consecutive zeros with a single zero
D) with the "::" symbol

Q5) Which type of IPv6 address is a global unicast address assigned to more than one interface? (Source: Defining IPv6 Addressing)

A) anycast
B) unicast
C) multicast
D) broadcast

Q6) Which address type from IPv4 was eliminated in IPv6? (Source: Defining IPv6 Addressing)

A) unicast
B) multicast
C) broadcast
D) everycast

Q7) Which statement is true about the EUI-64 address format of the system ID for stateless autoconfiguration used by Cisco? (Source: Implementing Dynamic IPv6 Addresses)

A) It is the MAC address plus the Site-Level Aggregator.
B) It is the MAC address plus the ISO OUI.
C) It expands the 48-bit MAC address to 64 bits by inserting FFFE into the middle 16 bits.
D) It does not follow IEEE standards for uniqueness of the address.
E) It is only used by Cisco.

Q8) Which two of the following are attributes of an IPv6 multicast address? (Choose two.) (Source: Implementing Dynamic IPv6 Addresses)

A) It begins with FFOO::/8.
B) It identifies a group of interfaces.
C) The multicast group ID consists of the last 120 bits of the address.
D) Scoping is defined by the Time to Live field.
E) Multicast addressing is used in IPv6 to define well-known duplicate addresses.

Q9) What is it called when an IPv6 router is involved in providing an IPv6 address to a requesting host? (Source: Implementing Dynamic IPv6 Addresses)

A) auto addressing
B) link local
C) IPv6 NAT
D) standard stateless autoconfiguration
E) DHCP autoconfiguration

Q10) Which two of the following are NOT IPv6 routing protocols? (Choose two.) (Source: Using IPv6 with OSPF and Other Routing Protocols)

A) IGRP6
B) OSPFv3
C) EIGRP for IPv6
D) RIPng
E) ODR
F) MP-BGP4

Q11) Which OSPFv3 address is the equivalent of 224.0.0.5 in OSPFv2? (Source: Using IPv6 with OSPF and Other Routing Protocols)

A) FF02::6
B) FF02::5
C) FF02::1
D) FF02::2

Q12) What are the two new LSAs in IPv6 OSPF? (Choose two.) (Source: Using IPv6 with OSPF and Other Routing Protocols)

A) interarea prefix LSA
B) interarea router LSA
C) link LSA
D) intra-area prefix LSA

Q13) What are the two most common IPv4-to-IPv6 transition techniques? (Choose two.) (Source: Using IPv6 with IPv4)

A) IPv6 NAT
B) dual stack
C) 6to4 tunnels
D) IPv6 mobile

Q14) Which is the global command that enables IPv6 or dual stack in a Cisco router? (Source: Using IPv6 with IPv4)

A) **ipv6 routing**
B) **ipv6 unicast-routing**
C) **ipv6 address**
D) **ipv6 dual stack**

Q15) Which two statements are true regarding dual stack? (Choose two.) (Source: Using IPv6 with IPv4)

A) A new API replaces gethostbyname and gethostbyaddr calls.
B) Tunneling is automatic.
C) Dual stack prefers IPv4 over IPv6.
D) IPv4 cannot be used while converting to IPv6.
E) The stack to use is chosen based on destination address.

Module Self-Check Answer Key

Q1)	B
Q2)	C
Q3)	A
Q4)	D
Q5)	A
Q6)	C
Q7)	C
Q8)	A, B
Q9)	D
Q10)	A, E
Q11)	B
Q12)	C, D
Q13)	B, C
Q14)	B
Q15)	A, E