



Americas Headquarters Cisco Systems, Inc. San Jose, CA	Asia Pacific Headquarters Cisco Systems (USA) Pte. Ltd. Singapore	Europe Headquarters Cisco Systems International BV Amsterdam, The Netherlands
---	--	---

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

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the United States and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

DISCLAIMER WARRANTY: THIS CONTENT IS BEING PROVIDED "AS IS" AND AS SUCH MAY INCLUDE TYPOGRAPHICAL, GRAPHICS, OR FORMATTING ERRORS. 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.

© 2014 Cisco Systems, Inc.

Table of Contents

Module 5: Routing Protocol Tools and Route Manipulation.....	1
Lesson 1: Introducing Routing Protocol Tools	3
Routing Protocol Tools Overview.....	5
Example: OSPF Filtering	7
Example: BGP Filtering.....	8
Route Filtering Tools.....	10
Typical Filtering Objectives in OSPF.....	11
Typical Filtering Objectives in IS-IS.....	12
Typical Filtering Objectives in BGP	13
Typical Routing Objectives.....	14
Prefix Lists Overview	16
Prefix List Syntax in Cisco IOS or IOS XE Software	17
Prefix List Example: Match Any Host Route	18
Prefix List Example: Match Default Route	19
Prefix List Example: Match All Routes	20
Prefix List Example: Match Small Routes	21
Prefix List Example: Match Core Loopbacks	22
Prefix List Example: Match Private Networks.....	23
Prefix List Syntax in Cisco IOS XR Software	24
Prefix List Examples in Cisco IOS XR Software	25
AS Path-Based Filtering	26
AS Path Access List Syntax in Cisco IOS or IOS XE Software.....	28
Regular Expressions, Special Characters.....	29
Commonly Used Regular Expressions.....	31
AS Path Access List Example: Permit All Routes.....	33
Example of AS Path Access List: Permit Local Routes	34
Example of AS Path Access List: Permit Routes From a Neighbor	35
Example of AS Path Access List: Permit Routes Originating in a Specific AS	36
Example of AS Path Access List: Permit Neighboring Local Routes.....	37
Example of AS Path Access List: Allow AS Path Prepending.....	38
Summary.....	39
Lesson 2: Introducing Route Maps and Routing Policy Language.....	41
Route Maps Overview	43
Route Map Processing	44
Route Maps Syntax	45
Example of Route Maps	47
Routing Policy Language.....	49
RPL Example	51
RPL Pass and Drop Actions	52
RPL Conditions	53
RPL Operators.....	54
RPL Boolean Operators	55
RPL Nesting.....	57
RPL Setting Attributes and Parameters.....	58
RPL Setting BGP Attributes and Parameters.....	61
Example: Setting BGP Attributes and Parameters.....	64
RPL Setting OSPF and IS-IS Parameters.....	66
RPL Parameterization.....	68
Applying Routing Policies	71

Maintaining Routing Policies	75
Value Sets	78
AS Path Sets	80
Standard Community Sets	83
Prefix Sets	87
Monitoring Routing Policies	88
Testing Routing Policies	90
Translating Route Maps to Routing Policies	91
Summary	93
Lesson 3: Implementing Route Redistribution	95
The Need for Redistribution	97
Route Redistribution	100
Using Seed Metrics in Route Redistribution	101
Default Seed Metrics in Route Redistribution	103
One-Point Route Redistribution	104
Multipoint Route Redistribution	105
Route Redistribution Techniques	107
Redistribution Implementation Considerations	109
Implement Redistribution	113
Redistribution into OSPF	114
Redistribution into IS-IS	116
Redistribution into BGP	117
Administrative Distance	118
Modifying OSPF Administrative Distance	119
Modifying IS-IS and BGP Administrative Distance	120
Example: Redistribution Using Administrative Distance	121
Redistribution to Prevent Routing Loops	126
Summary	128
Lesson 4: Influencing BGP Route Selection	129
BGP Weight	131
Configuring Per-Neighbor Weights	133
Changing Weights with RPLs or Route Maps	134
BGP Weight Attachment Points	135
BGP Local Preference	136
Changing Local Preference	138
Monitoring Local Preference	139
AS Path Prepending	140
AS Path Prepending Design Considerations	145
Configuring AS Path Prepending	147
Monitoring AS Path Prepending	148
AS Path Filtering Concerns: AS Path Prepending	149
BGP Multi-Exit Discriminators	151
Changing the MED	153
Monitoring the MED	155
BGP Communities	156
Steps for Designing a BGP solution around BGP communities	158
Configuring BGP Communities	160
BGP Named Community Lists	164
BGP Support for Sequenced Entries in Extended Community Lists	165
Matching BGP Communities	166
Monitoring BGP Communities	167
BGP Community Examples	169
Summary	175

Lesson 5: Module Summary	177
Lesson 6: Module Self-Check	179
Module Self-Check Answers.....	185
Answer Key	185
Glossary.....	187

Module 5: Routing Protocol Tools and Route Manipulation

Introduction

This module describes tools that are used for routing protocol manipulation, route redistribution, and BGP route selection. This module also presents the mechanisms that are available to service provider administrators; these mechanisms, in combination with routing protocols, are used to filter routing information or to implement desired routing policies.

Upon completing this module, you will be able to:

- Identify the main characteristics of routing protocols that are used in service provider environments
- Describe route maps and routing policy language
- Explain what route redistribution is, why it is necessary, and how to configure route redistribution
- Complete the correct BGP configuration to successfully connect the service provider network to the customer or upstream service provider

Lesson 1: Introducing Routing Protocol Tools

Overview

The lesson focuses on the mechanisms that are available, in combination with routing protocols, to filter routing information or to implement desired routing policies using Cisco IOS/IOS XE and Cisco IOS XR routers. The lesson describes prefix lists and AS path access lists.

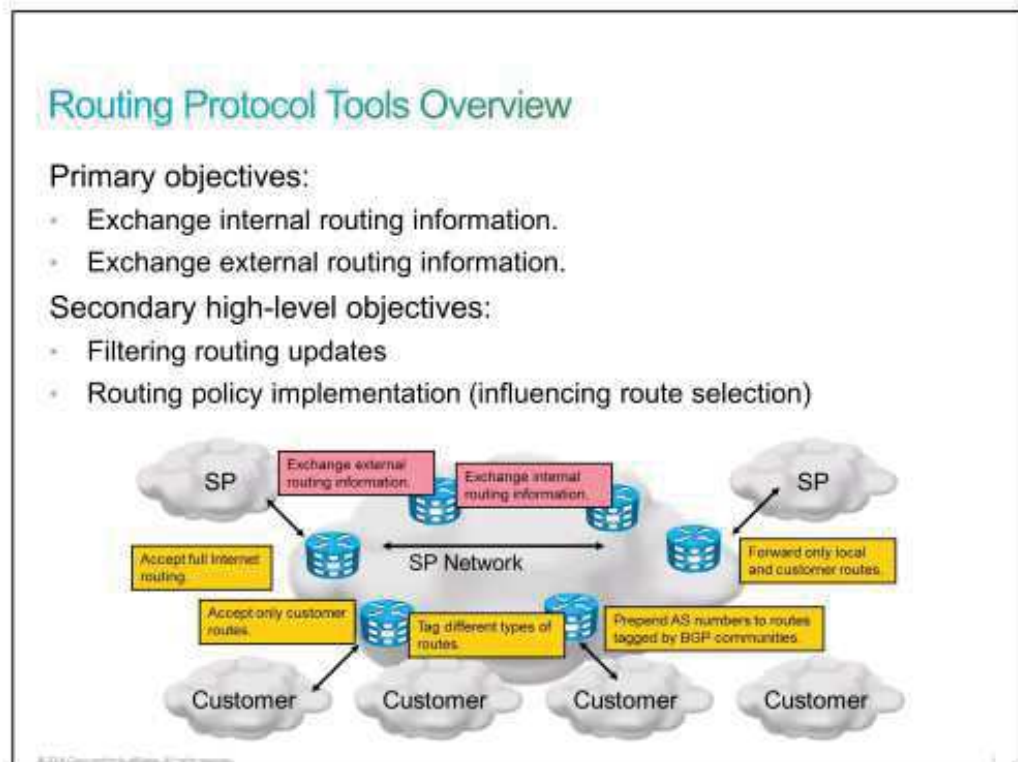
Upon completing this lesson, you will be able to:

- Describe the characteristics and requirements for routing policies in service-provider environments
- Provide examples of OSPF filtering
- Provide examples of BGP filtering
- Describe the most common tools used to implement route filtering
- Describe typical filtering objectives in OSPF networks
- Describe typical filtering objectives in IS-IS networks
- Describe typical filtering objectives in BGP networks
- Describe the routing policy requirements that are typically implemented using BGP
- Provide an overview of prefix lists
- Describe the syntax for prefix lists in Cisco IOS and IOS XE software
- Show an example of matching any host route in Cisco IOS and IOS XE software
- Show an example of matching a default route in Cisco IOS and IOS XE software
- Show an example of matching all routes in Cisco IOS and IOS XE software
- Show an example of matching all small prefixes in Cisco IOS and IOS XE software
- Show an example of matching all host routes in a range in Cisco IOS and IOS XE software

- Show an example of matching private networks in Cisco IOS and IOS XE software
- Describe the syntax for prefix lists in Cisco IOS XR software
- Show examples of prefix lists in Cisco IOS XR Software
- Describe the characteristics and usage scenarios for AS path-based filtering in service provider environments
- Describe the syntax for the AS path access list command in Cisco IOS and IOS XE software
- Describe the special characters used to create regular expressions
- List commonly used regular expressions
- Show an example AS path access list to permit all routes in Cisco IOS and IOS XE software
- Show an example AS path access list to permit local routes in Cisco IOS and IOS XE software
- Show an example AS path access list to permit routes from a specific neighbor in Cisco IOS and IOS XE software
- Show an example AS path access list to permit routes from a specific AS in Cisco IOS and IOS XE software
- Show an example AS path access list to permit routes that originate in a neighboring AS in Cisco IOS and IOS XE software
- Show an example AS path access list to permit local prefixes with AS path prepending in Cisco IOS and IOS XE software

Routing Protocol Tools Overview

Various actions can be performed on routing updates in a typical service provider environment.



The actions can be divided into two main categories:

- Exchanging routing information (the primary objective of routing protocols)
- Implementing a routing policy and filtering routing information

To exchange routing information, a typical service provider would use two routing protocols:

- An IGP such as OSPF, or IS-IS to exchange local routing information.
- BGP to exchange external routing information (for example, customer routing information and complete Internet routing information from other service providers).

BGP will always be combined with advanced filtering and policy mechanisms for security and performance reasons. This lesson will discuss various mechanisms that can be used for filtering or for routing policy implementation.

Typical Filtering Objectives

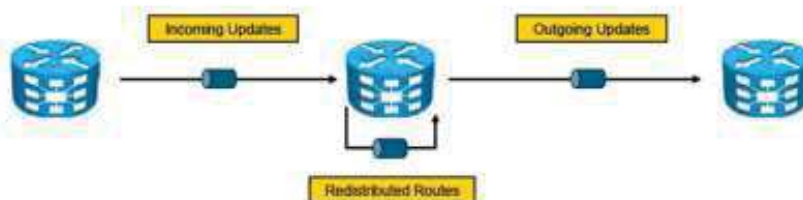
Typical Filtering Objectives

Filter:

- Incoming updates
- Outgoing updates
- Redistributed routes from other routing protocols

Filter based on:

- Prefix and prefix length (subnet mask)
- Update parameters (routing-protocol-specific)



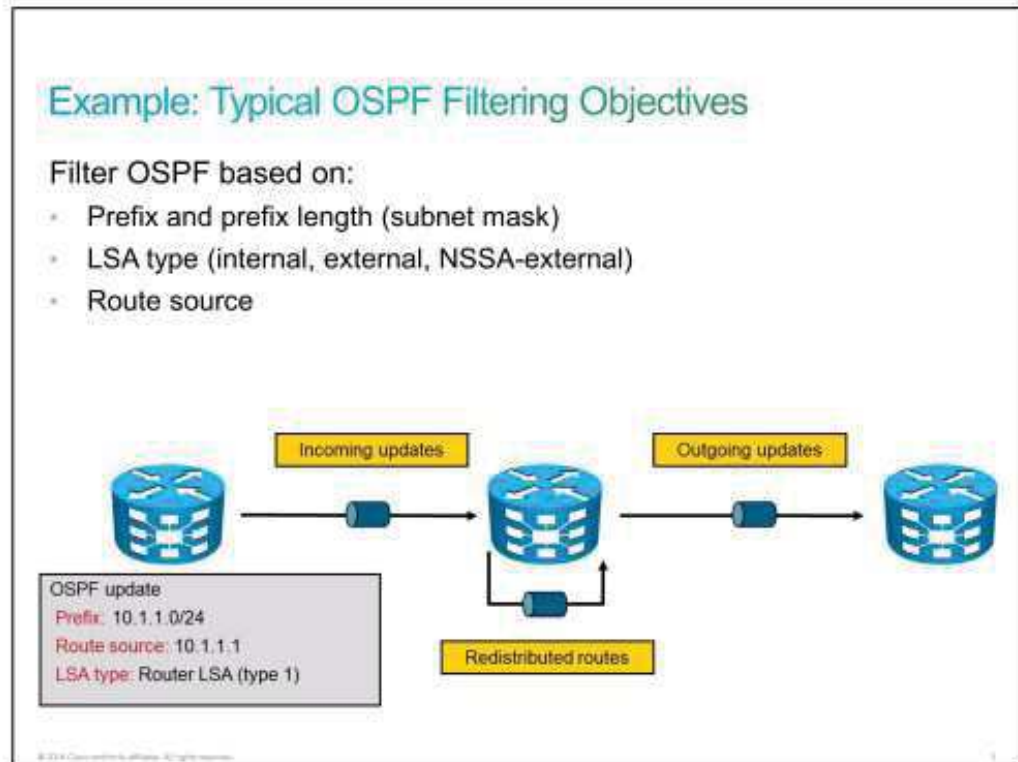
Filtering of routing information can be performed in these three locations:

- On incoming updates, as they are received from a neighboring router
- On outgoing updates, before they are sent to a neighboring router
- On redistribution from another routing protocol, including static and connected route redistribution

Filtering can be based on the prefix itself or based on some other parameter that is available in the routing protocol or source routing protocol (when you are doing redistribution).

Example: OSPF Filtering

An OSPF update that carries information can be used for filtering purposes.



There are many ways to filter OSPF updates:

- Prefix and prefix length
- Route source (that is, the advertising IP address of the router)
- OSPF LSA type

Filtering of redistributed routes can be performed on any router that effectively becomes an ASBR, while regular filtering of OSPF updates can only be performed on ABRs for routes forwarded from one area into another.

Example: BGP Filtering

As with OSPF, a BGP update can be used for filtering purposes.

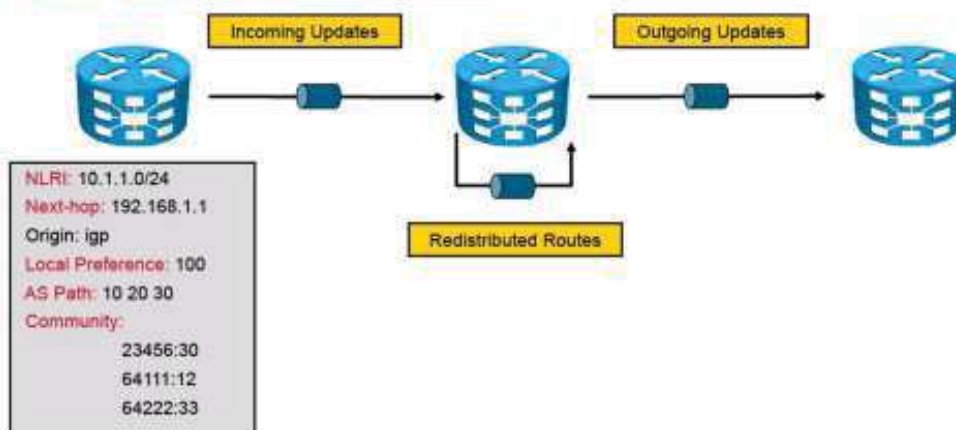
Example: BGP Filtering

Example: typical BGP filtering objectives

Filter BGP based on:

- Prefix and prefix length (subnet mask)
- Next-hop address
- Route source address
- AS path attribute
- BGP community and BGP extended-community attributes
- Local preference attribute

Example: BGP Filtering (Cont.)



The figure illustrates a BGP update, that has a much richer metric, that is, collection of BGP attributes that can also be used for filtering purposes.

Filtering of BGP updates can be performed based on these attributes:

- Prefix and prefix length (subnet mask) found in the BGP NLRI
- Next-hop address found in the BGP next-hop attribute
- Route source address (that is, the IP address of the neighbor)
- AS path attribute
- BGP community and BGP extended-community attributes
- Local preference attribute

Route Filtering Tools

You can use prefix lists, AS path access lists, route maps, or routing policies as a filtering tool.

Route-Filtering Tools

Prefix lists:

- Used for prefix-based filtering or matching of routes
- Can be used to match on the prefix, route source, or next-hop address

AS path access lists:

- Used in BGP for filtering or route matching based on the BGP AS path attribute

Route maps:

- Primarily used to implement complex routing policies
- Can also be used as a powerful filtering tool

Routing policy language:

- Replaces route maps in Cisco IOS XR Software
- Feature-rich language for complex routing policies

© 2014 Cisco and/or its affiliates. All rights reserved.

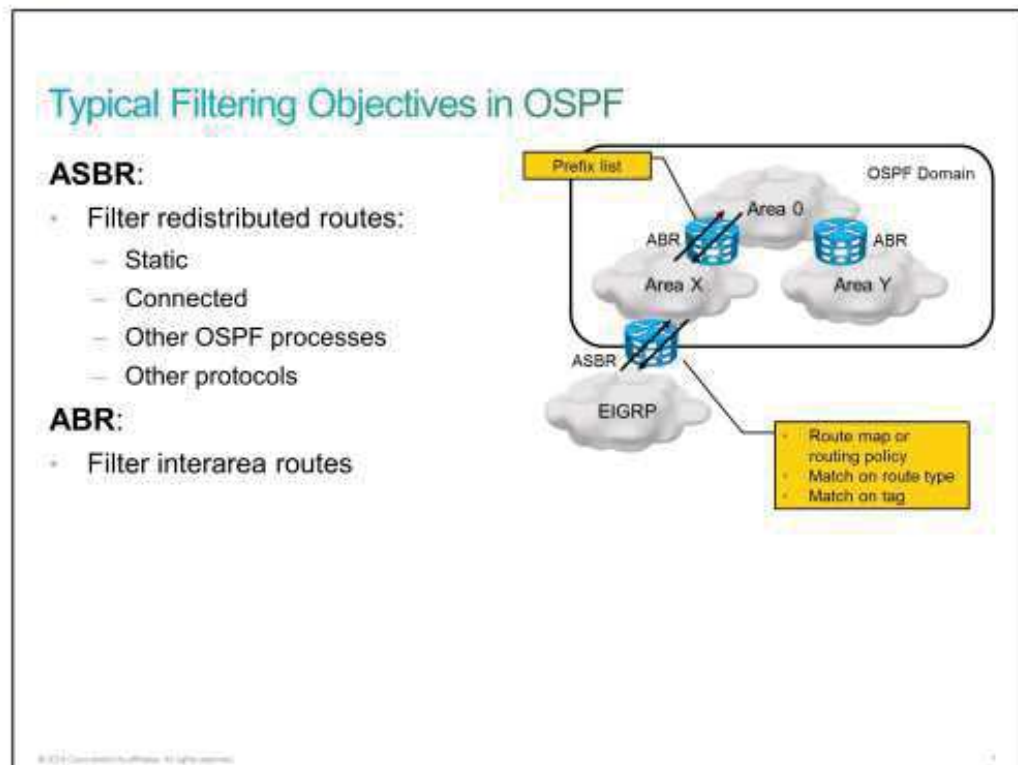
5

The following tools are most commonly used to implement filtering and routing policies in Cisco IOS or IOS XE or Cisco IOS XR Software or both:

- **Prefix lists** can be used to implement filtering or matching routing updates that are based on [IP address](#) or [IP](#) network information; some examples are a prefix, a next-hop address, or the address of a neighbor. Prefix lists are available in Cisco IOS or IOS XE Software. Prefix lists are also available in Cisco IOS XR Software with slight differences.
- **AS path access lists** can be used with BGP to implement filtering or matching of routing updates that are based on the contents of the AS path attribute. A regular expression is used to process the AS path as a string of characters. AS path access lists are only available in Cisco IOS or IOS XE Software. Cisco IOS XR Software matches AS path attributes directly in routing policies.
- **Route maps** are primarily used to implement routing policies that can modify routing protocol parameters as well as perform filtering. Route maps are only available in Cisco IOS or IOS XE Software.
- **Routing policies** are a more powerful and flexible version of the route maps that are available in Cisco IOS XR Software.

Typical Filtering Objectives in OSPF

With OSPF, only ABRs or ASBRs can perform route filtering.



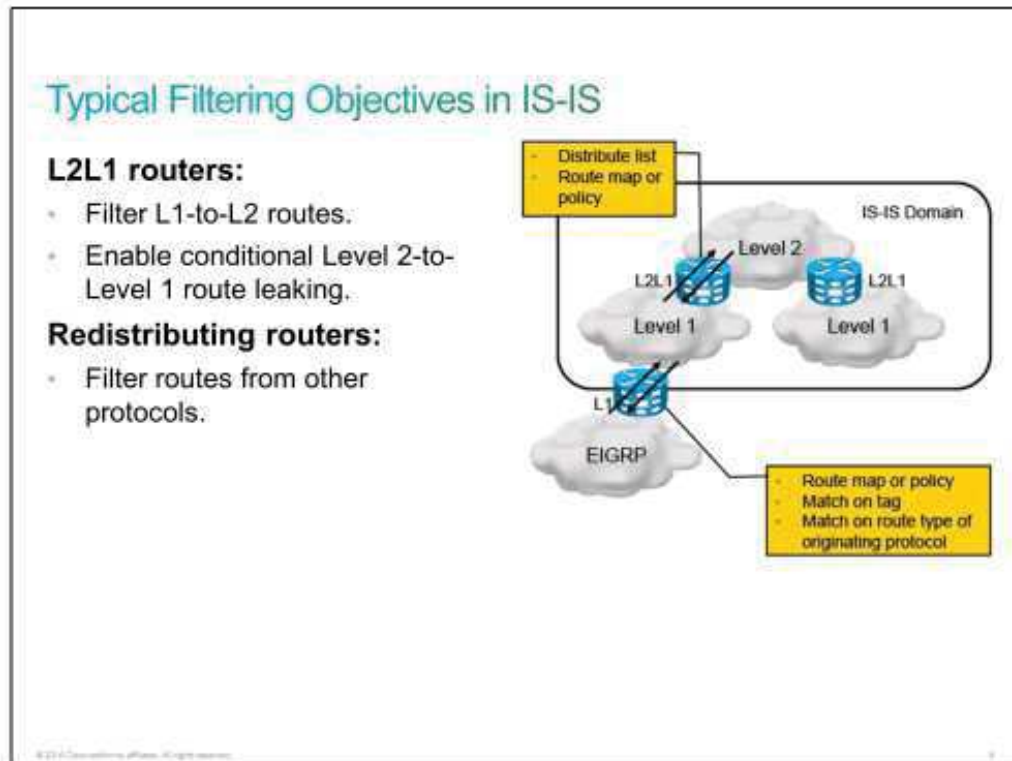
The figure illustrates an OSPF domain that is using multiple OSPF areas and a connection to an external EIGRP AS.

ASBRs perform routing exchange with other routing protocols or OSPF domains by performing redistribution of routing information. ASBRs can filter redistributed routes by using route maps or routing policies from connected routes, static routes, other OSPF processes, IS-IS, EIGRP, RIP, or even BGP (not recommended).

ABRs perform an exchange of routing information between OSPF areas within the same OSPF domain, according to OSPF rules. Prefix lists can be used to control the exchange of routing information between areas.

Typical Filtering Objectives in IS-IS

Any IS-IS router can perform redistribution from other routing protocols, using a route map or routing policy to control the redistribution of routes.

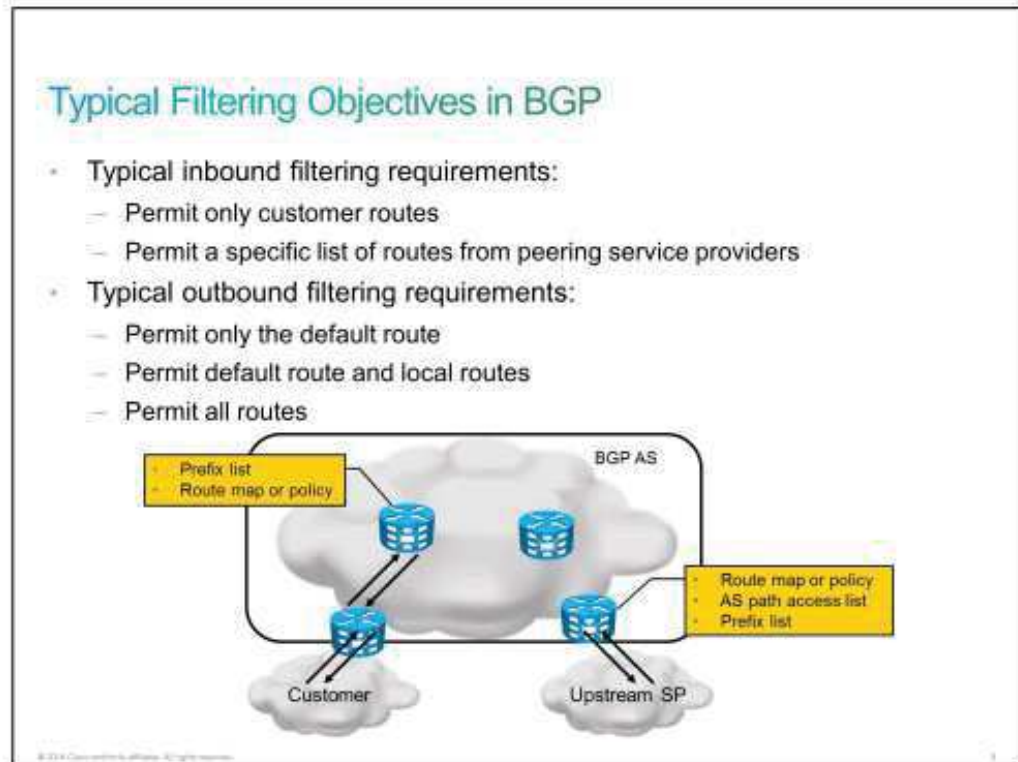


The figure illustrates an IS-IS domain that is using multiple IS-IS levels and a connection to an external EIGRP AS.

L2L1 routers (like ABRs) perform routing exchange for both IS-IS levels. Prefix lists, route maps, or routing policies can be used to filter an exchange of routing information between IS-IS levels. Route leaking can also be used to control the distribution of Level 2 routes into Level 1.

Typical Filtering Objectives in BGP

With BGP, several inbound and outbound filtering requirements exist.



The figure illustrates a sample BGP AS using BGP route reflectors (to reduce the full-mesh IBGP requirements) and edge BGP routers to implement routing for external destinations.

Inbound filtering can depend on the type of neighboring AS:

- Permit only customer routes for end customers
- Permit a specific list of routes from subordinate service providers, service providers that are peering at an exchange point
- Permit the complete Internet routing information from upstream service providers

These are **outbound** filtering requirements:

- Permit only the default route (for example, single-homed customers do not require more specific information; most single-homed customers do not even require a routing protocol)
- Permit default route and local routes (for example, multihomed customers that are using this service provider as a backup provider, but still want to access local destinations directly)
- Permit all routes (for example, multihomed customers that require complete Internet routing information)

Typical Routing Objectives

Routing policies are most commonly implemented for external routing information, using BGP.

Typical Routing Objectives

Complex routing policies are most often implemented using BGP.

Influencing route selection for:

- Outgoing traffic
- Incoming traffic

Routing decision influenced:

- Locally
- Remotely (e.g. by customer or downstream service provider)

A routing policy can address the outgoing path or the return path. Additionally, BGP can be used to implement a policy locally within an AS, or used to have a neighboring AS influence the route selection in an AS. (One example is using AS path prepending, or signaling a policy using BGP communities that are translated to local preference in an AS).

Typical Routing Objectives in BGP

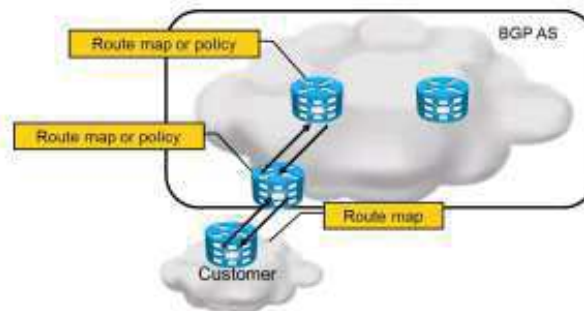
Typical Routing Objectives in BGP

Customer selecting primary or backup ISP:

- AS path prepending by customer
- BGP community sent by customer

Policy implemented by service provider:

- Setting local preference
- Translating BGP community to local preference



A routing policy is always implemented by using route maps in the Cisco IOS or IOS XE Software or routing policies in Cisco IOS XR Software. Implementing policies that are consistent across the entire AS (that is, implementing policies on edge routers) is recommended.

The figure lists some commonly implemented policies in service provider environments:

- Customers often use AS path prepending to artificially lengthen the AS path attribute, thus making it less desirable (that is, to signal that this service provider is the backup service provider).
- Customers can alternatively signal their service provider preference by using some BGP communities that are offered by a service provider. The service provider will then translate the BGP communities received from the customers to some other BGP attribute (for example, AS path prepending or local preference) to influence the outbound traffic to the customers.
- Service providers can use the BGP local preference attribute to influence route selection internally, within the service provider AS (for example, to select preferred upstream service providers).

Prefix Lists Overview

Prefix lists are designed to simplify the filtering of routing updates.

Prefix List Overview

Prefix lists characteristics:

- Designed for route filtering or matching
- Replaces access lists that were designed for packet filtering or matching
- Available in Cisco IOS or IOS XE and Cisco IOS XR Software with slight differences

Prefix lists are available in Cisco IOS, IOS XE (for the [ASR](#) router family), and IOS XR Software (with some slight differences).

Prefix List Syntax in Cisco IOS or IOS XE Software

Like all other named objects in Cisco IOS or IOS XE and Cisco IOS XR Software, each prefix list is identified by using a case-sensitive name.

Prefix List Syntax in Cisco IOS or IOS XE Software

Cisco IOS or IOS XE Software prefix lists syntax:

- Each prefix list is identified using a case-sensitive *name*.
- Each prefix list can have one or more lines.
- Edit and order prefix list entries by using line numbers.
- The *network/length* pair identifies the bits in prefixes to match.
- The **ge** and **le** operators identify the length of prefixes to match:
 - **le** :“less or equal” matches any prefix that is shorter or equal in length to the specified value.
 - **ge** :“greater or equal” matches any prefix that is longer or equal in length to the specified value.
 - **ge x le x** :“equal” (There is no “eq” operator in Cisco IOS or IOS XE Software.)

```
Router(config)# ip prefix-list name [seq num] {deny|permit} net/length [ge len]
[le len]
```

© 2014 Cisco Systems, Inc. All rights reserved.

13

A prefix list can have multiple lines that are ordered using line numbers.

The *network/length* pair identifies the bits in prefixes to match. The **ge** and **le** operators identify the length of the prefixes to match. A combination of both operators can be used to match a range of prefix lengths or a specific length: **ge x le x** means “equal,” since there is no **eq** operator in Cisco IOS or IOS XE Software.

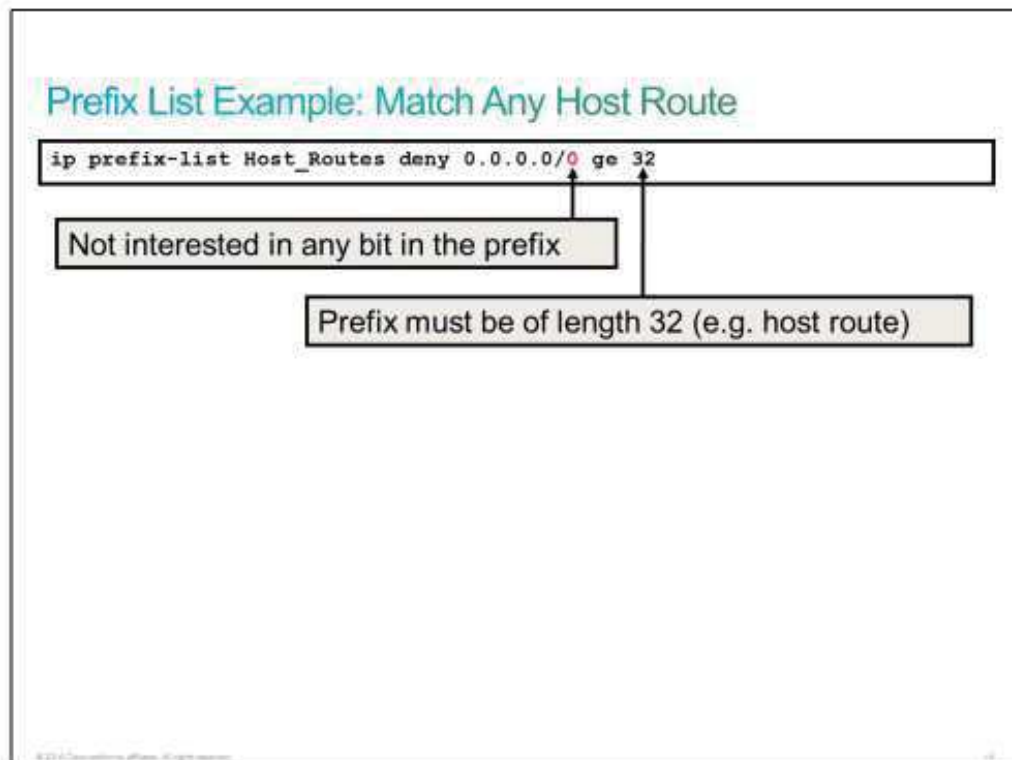
Full syntax:

```
ip prefix-list [list-name [seq number] {deny | permit} network/length [ge ge-length]
[le le-length] | description description [sequence-number]
```

For IPv6, the **ipv6 prefix-list name** command is used to match IPv6 routing information.

Prefix List Example: Match Any Host Route

Host routes are often filtered out to minimize the size of the routing table. An example of matching any host route in Cisco IOS and IOS XE software is shown below.

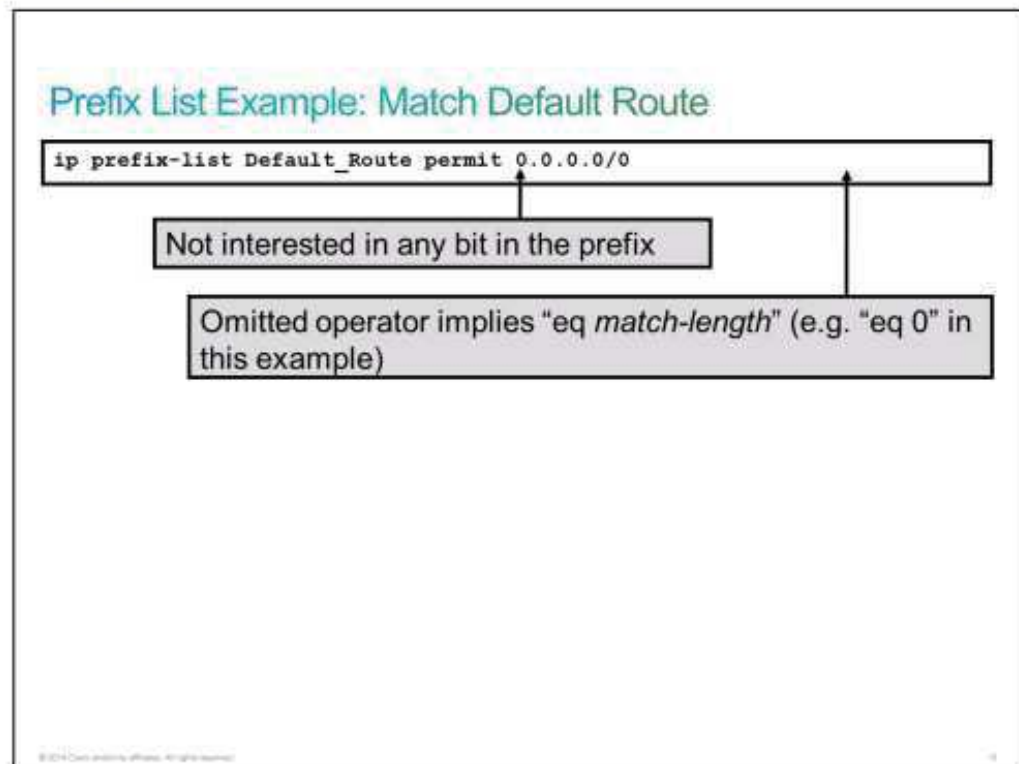


The sample prefix list shows how to match any host route:

- The “0” in the prefix length indicates that there is no interest in any bit in the prefix itself.
- The “ge 32” indicates that the length of the prefix (that is, subnet mask) must be 32 (that is, 255.255.255.255), thus matching host routes.

Prefix List Example: Match Default Route

Single-homed customers running [BGP](#) or multi-homed customers that do not require full Internet routing should receive only the default route. An example of matching a default route in Cisco IOS and IOS XE software is shown below.



The sample prefix list shows how to match a default route:

- The "0" in the prefix length indicates that there is no interest in any bit in the prefix itself.
- The omitted operator indicates that the prefix length should be the same as the number of bits in the prefix that it should be matched with (that is, 0, that is the length of the subnet mask in a default route).

Prefix List Example: Match All Routes

There is no keyword **any** as in access lists. An example of matching all routes in Cisco IOS and IOS XE software is shown below.

Prefix List Example: Match All Routes

Use this example to match any route.

```
ip prefix-list All_Prefixes permit 0.0a.0.0/0 le 32
```

Not interested in any bit in the prefix

Prefix can be of any length from 0 to 32 (e.g. any route)

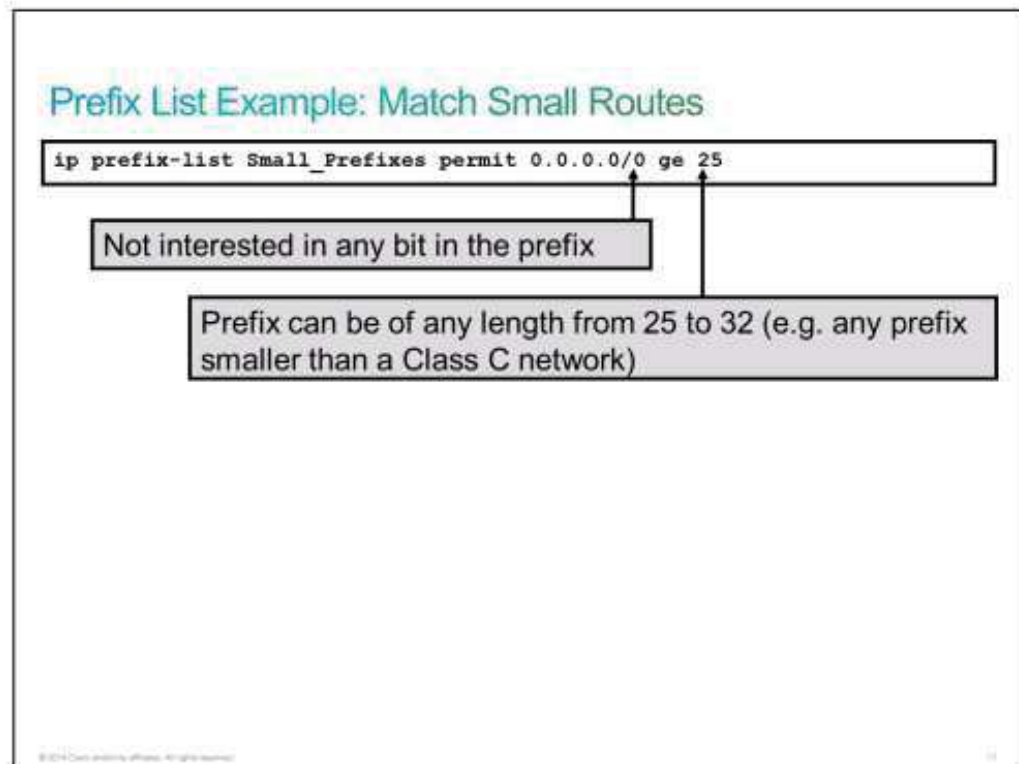
© 2014 Cisco Systems, Inc. All rights reserved.

The sample prefix list shows how to match any route:

- The “0” in the prefix length indicates that there is no interest in any bit in the prefix itself.
- The “le 32” operator indicates that the prefix length can be anything from 0 to 32, thus matching any subnet mask.

Prefix List Example: Match Small Routes

Small prefixes (for example, prefix length greater than 24) are sometimes filtered out to minimize the size of the full Internet routing table. An example of matching all small prefixes in Cisco IOS and IOS XE software is shown below.

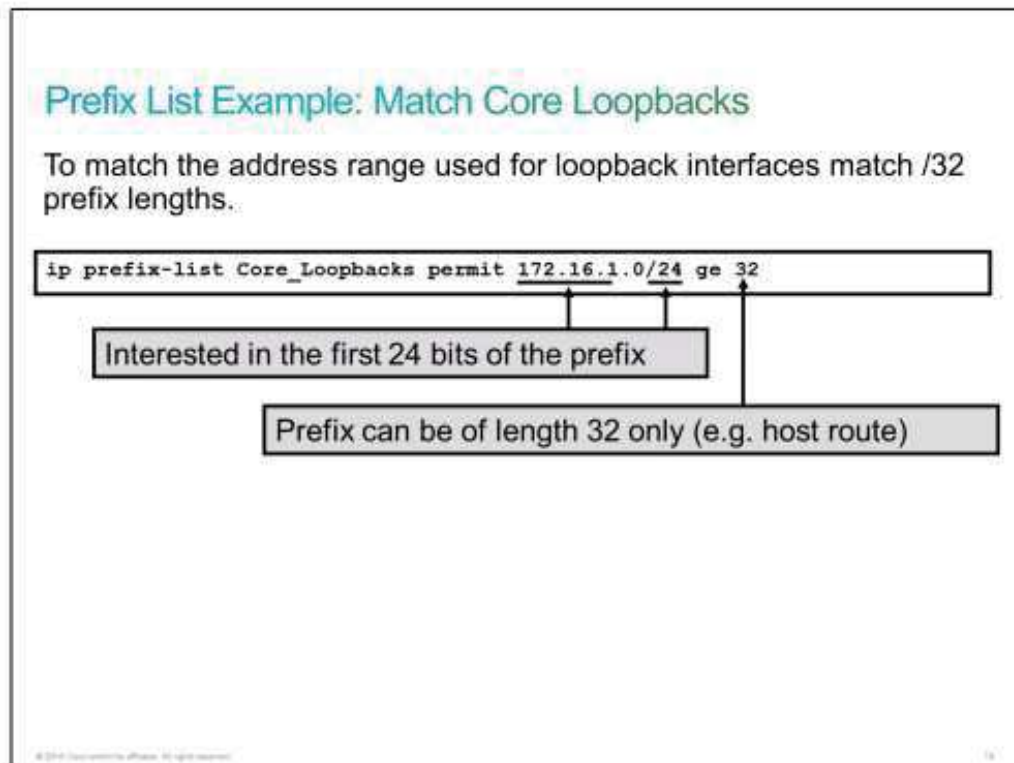


The sample prefix list shows how to match all small prefixes that have a prefix length greater than a Class C equivalent:

- The “0” in the prefix length indicates that there is no interest in any bit in the prefix itself.
- The “ge 25” operator indicates that the prefix length can be anything from 25 to 32, thus matching any subnet mask from 255.255.255.128 to 255.255.255.255.

Prefix List Example: Match Core Loopbacks

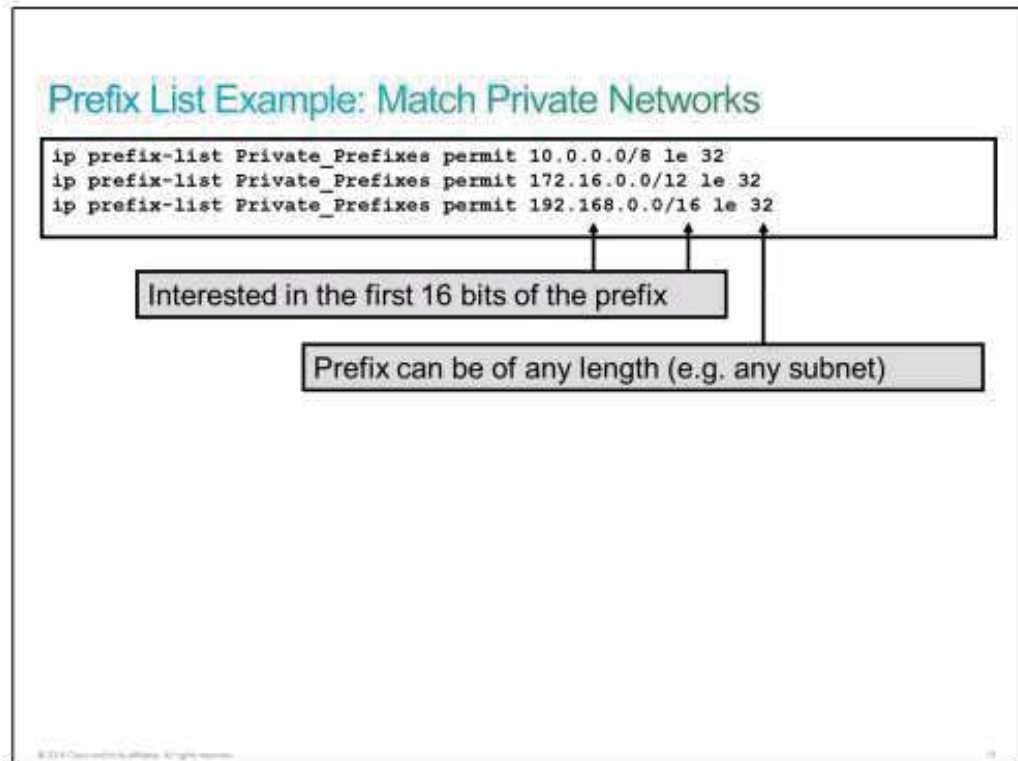
Host routes are sometimes matched (for example, loopback addresses). An example of matching all host routes in a range in Cisco IOS and IOS XE software is shown below.



The sample prefix list matches all host routes in a given range of prefixes (for example, 172.16.1.1/32, 172.16.1.2/32, and so on). This type of prefix list is useful for matching (for example, loopback addresses).

Prefix List Example: Match Private Networks

Private networks are always filtered out when sending updates to other autonomous systems. An example of matching private networks in Cisco IOS and IOS XE software is shown below.



The sample prefix list matches any network or subnet in the [RFC 1918](#) range of the [IP address](#) (that is, private address space). These private networks are typically filtered out on a routing exchange between autonomous systems.

The “le 32” is used whenever you are not interested in the size of the prefix (that is, to match any subnet).

Prefix List Syntax in Cisco IOS XR Software

Prefix List syntax in Cisco IOS XR Software is similar to Cisco IOS or IOS XE Software syntax, except that it is modular.

Prefix List Syntax in Cisco IOS XR Software

Cisco IOS XR Software prefix list syntax:

- Each prefix list is identified using a case-sensitive *name*.
- Edit and order prefix list using line numbers.
- The *network/length* pair identifies the bits in the prefix that must match.
- The **ge**, **le**, and **eq** operators identify the length of prefixes to match:
 - **le**: "less or equal" matches any prefix that is shorter or equal in length to the specified value.
 - **ge**: "greater or equal" matches any prefix that is longer or equal in length to the specified value.
 - **eq**: "equal" matches any prefix of the exact specified length.

```
RP/0/RP0/CPU0:CRS(config)# ipv4 prefix-list name [seq num] {deny | permit}  
network/length [ge len] [le len] [eq len]
```

© 2014 Cisco and/or its affiliates. All rights reserved.

Prefix list syntax in Cisco IOS XR Software is different from that in Cisco IOS or IOS XE Software, only in that it also implements the **eq** operator to match an exact prefix length.

For IPv6, the **ipv6 prefix-list** *name* command is used to match IPv6 routing information.

Prefix List Examples in Cisco IOS XR Software

Prefix lists examples are shown for Cisco IOS XR Software.

Prefix List Examples in Cisco IOS XR Software

Cisco IOS XR Software

```
ipv4 prefix-list Private_Prefixes
deny 10.0.0.0/8 le 32
deny 172.16.0.0/12 le 32
deny 192.168.0.0/16 le 32
permit 0.0.0.0/0 le 32
!
ipv4 prefix-list Core_Loopbacks
permit 172.16.1.0/24 eq 32
!
ipv4 prefix-list Host_Routes
permit 0.0.0.0/0 eq 32
!
ipv4 prefix-list Default_Route
permit 0.0.0.0/0
!
ipv4 prefix-list All_Prefixes
permit 0.0.0.0/0 le 32
!
ipv4 prefix-list Small_Prefixes
permit 0.0.0.0/0 ge 25
!
```

The **Private_Prefixes** prefix list shows how to filter out all [RFC 1918](#) prefixes. These types of filters are commonly used on incoming and outgoing updates on [EBGP](#) sessions.

The **Core_Loopbacks** prefix list illustrates how to match host routes that can be used to match loopback addresses from a given address range.

The **Host_Routes** prefix list illustrates how to match any host route.

The **Default_Route** prefix list illustrates how to match only the default route.

The **All_Prefixes** shows how to match any (all) prefixes. (The prefix list line equals the **any** keyword that is used in access lists to match any network).

The **Small_Prefixes** illustrates how to filter out all small prefixes that have a prefix length equal to or greater than 25 (small prefixes).

AS Path-Based Filtering

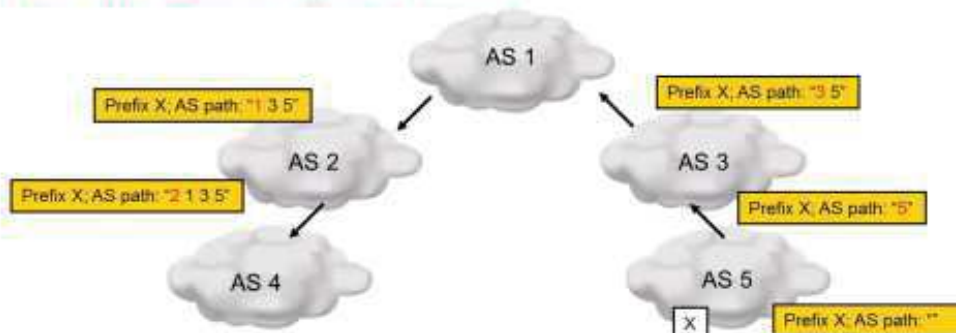
An AS path access list can be used to identify various updates that are based on the characteristics of their AS path attribute. Regular expressions are used to process AS path attributes.

AS Path-Based Filtering

AS path-based filtering characteristics:

- BGP uses autonomous systems to identify the origin and path of a prefix.
- Each path is identified using a sequence of AS numbers.
- AS path attribute is used to carry the AS path in BGP updates.
- Each egress BGP router prepends its own AS number to the AS path attribute.
- AS path access lists are used to match prefixes based on AS path characteristics.

AS Path-Based Filtering (Cont.)



The figure illustrates the automatic prepending that is done by all egress routers when they send updates to a neighboring AS. It shows that the first number in the AS path is always the number of the neighboring AS from which the update was received. The last number in the AS path is the number of the originating AS.

AS Path Access List Syntax in Cisco IOS or IOS XE Software

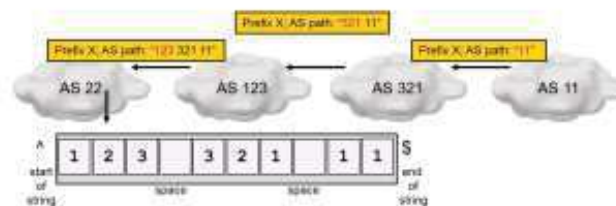
Each AS path access list is identified by using a unique number in the range from 1 to 500. Regular expressions are used to match prefixes that are based on contents of the AS path attribute that is converted to a string of characters.

AS Path Access List Syntax in Cisco IOS or IOS XE Software

AS path access list syntax in Cisco IOS or IOS XE:

- Each AS path access list is identified by using a unique number.
- Regular expressions are used to match prefixes based on the contents of the AS path attribute.
- The AS path is processed as a string of characters.

```
Router(config)# ip as-path access-list acl-number {permit | deny} regexp
```



The figure illustrates an AS path attribute as seen in AS 22. The AS path is converted to a string of characters that starts with character "1" and also ends with character "1" in this AS path example. Regular expressions must be written to take into account that the typical need is to identify AS numbers and not individual characters.

Regular Expressions, Special Characters

Listed here are the special characters used to create regular expressions.

Regular Expressions, Special Characters

Character	Description
^	matches the start of AS path (e.g. <code>^20_</code>)
\$	matches the end of AS path (e.g. <code>^20\$</code>)
_	matches any delimiter (start, end, or space; e.g. <code>_20_</code>)
.	matches any single character
*	matches preceding character any number of times including zero (e.g. <code>.*^20(_20)*\$</code>)
+	matches preceding character once or more times (e.g. <code>^[0-9]+\$</code>)
?	matches preceding character zero or one time (e.g. <code>^20(_20)?\$</code>)

Regular Expressions, Special Characters (Cont.)

Character	Description
 	logical OR operator (e.g. <code>_100_ _200_</code>)
()	groups characters for precedence or to capture matched values into \n (e.g. <code>_100_(200 300)_</code>)
[range]	matches a single character from the defined range of characters (e.g. <code>"[0-9]"</code> , <code>"[13579]"</code>)
\n	matches again what was found within the n-th pair of parentheses (e.g. <code>"([0-9]+)(_1)*"</code>)
\X	removes the special meaning of character X (e.g. <code>"\" or "\")</code>)

RegEx are strings of special characters that can be used to filter character patterns. They are made up of symbols, numbers, and letters. They are used in **show** commands as well as in BGP (AS-PATH) ACLs to match a BGP route, based on the information that is on its AS-PATH.

Note	The text match is sensitive to capitals and spacing. This becomes important when referencing interfaces, that are always capitalized.
-------------	---

Commonly Used Regular Expressions

Listed here is the list of commonly used regular expressions.

Commonly Used Regular Expressions	
Regular Expression	Description
<code>^\$</code>	matches locally originated prefixes
<code>^number\$</code>	matches prefixes originating in the specified neighboring AS
<code>_number\$</code>	matches prefixes originating in the specified AS
<code>^number_</code>	matches prefixes learned through the specified neighboring AS
<code>^([0-9]+)(_1)*\$</code>	matches prefixes originating in any neighboring AS and allowing prepending
<code>.*</code>	matches all prefixes (e.g. "any")
<code>.</code>	matches nonlocal prefixes (e.g. all except empty AS path)

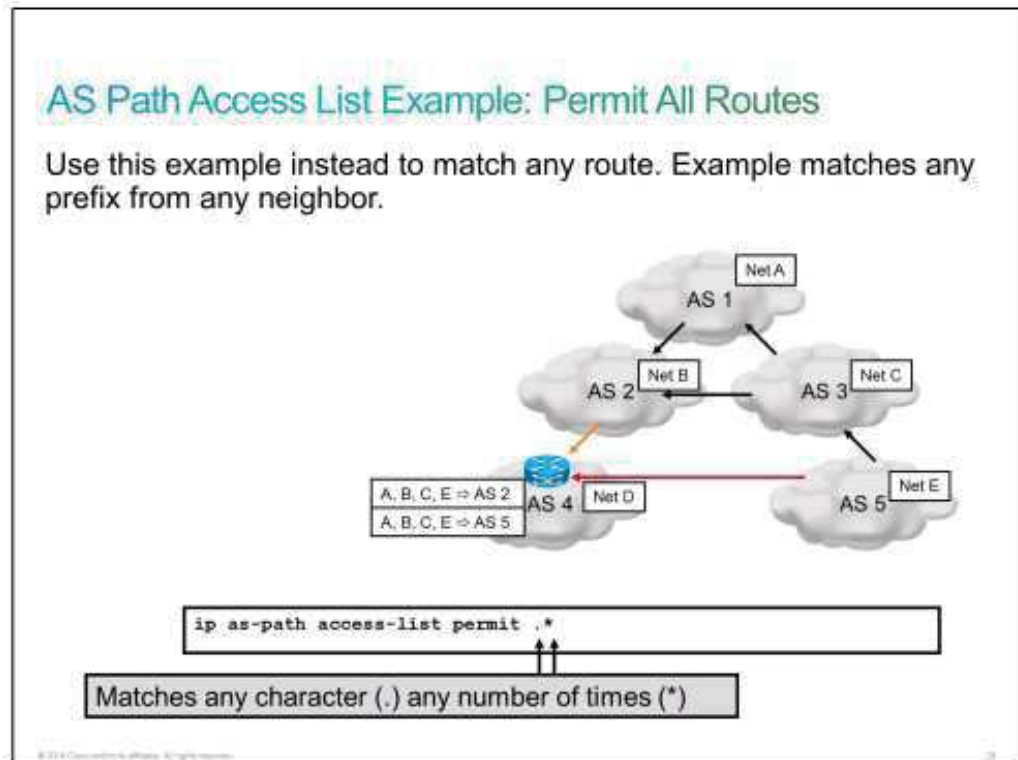
The table shows some examples of commonly used regular expressions.

Commonly Used Regular Expressions

Regular Expression	Description
<code>^\$</code>	Matches all local routes (local routes have an empty AS path attribute)
<code>^10\$</code>	Matches any route originating in a neighboring AS 10
<code>_20\$</code>	Matches any route originating in AS 20
<code>^10_</code>	Matches any route received from a neighboring AS 10
<code>{[0-9]+}(_\1)*</code>	Matches any AS number, which can optionally repeat any number of times (that is, prepending). \1 references whatever is matched in the first pair of parentheses.
<code>.</code>	Matches any character any number of times. This regular expression is used to match any prefix.
<code>.</code>	Matches any single character
<code>_20_</code>	Matches any route originating or passing through AS 20
<code>[0-9]+</code>	Matches any AS number from 0 to 65335 (maximum range in BGP)
<code>[13579]\$</code>	Matches routes originating in odd-numbered autonomous systems
<code>[02468]\$</code>	Matches routes originating in even-numbered autonomous systems

AS Path Access List Example: Permit All Routes

There is no keyword **any**, as in access lists. An example of an AS path access list to permit all routes in Cisco IOS and IOS XE software is shown below.

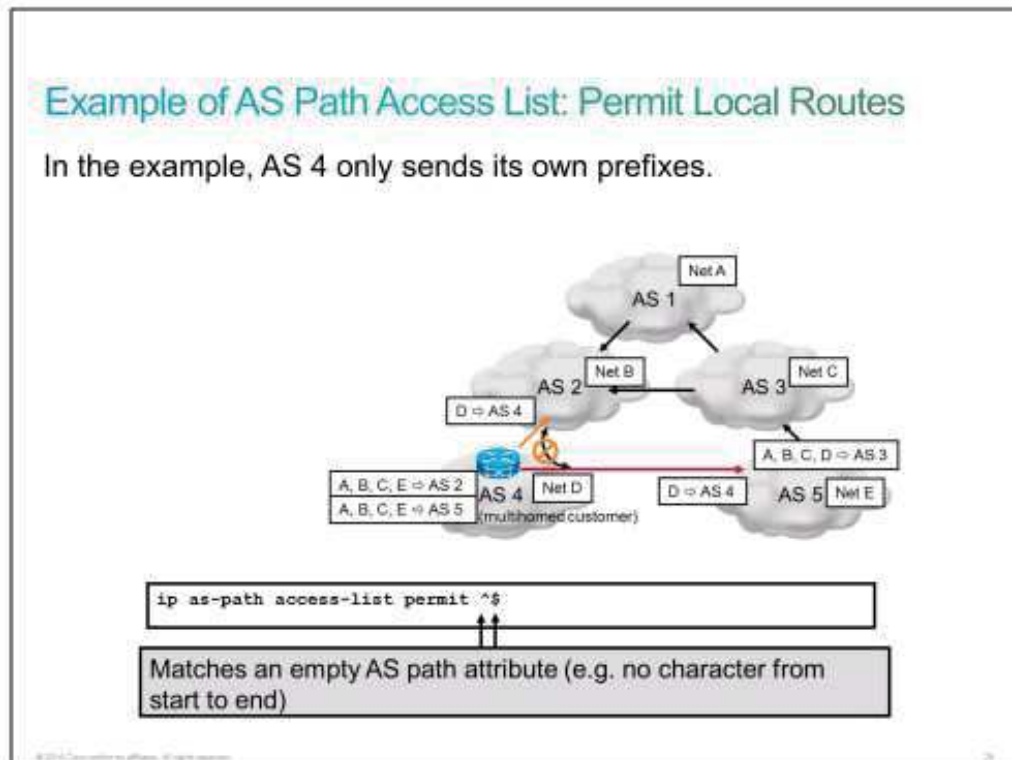


The sample regular expression (**ip as-path access-list permit .***) matches any character any number of times. This AS path access list entry is used to permit any route (that is, the equivalent of the **any** keyword in access lists).

The figure illustrates five autonomous systems, each represented by one prefix that it advertises. AS 1, for example, advertises prefix “A” which can be learned by AS 4 from AS 2, AS 5, or both. If this filter is applied in AS 4 to incoming updates from AS 2 or AS 5, all routes would be accepted.

Example of AS Path Access List: Permit Local Routes

Locally originated routes have an empty AS path attribute. Customers typically use this outbound filter to announce their prefixes to their service providers. A sample AS path access list to permit local routes in Cisco IOS and IOS XE software is shown here.

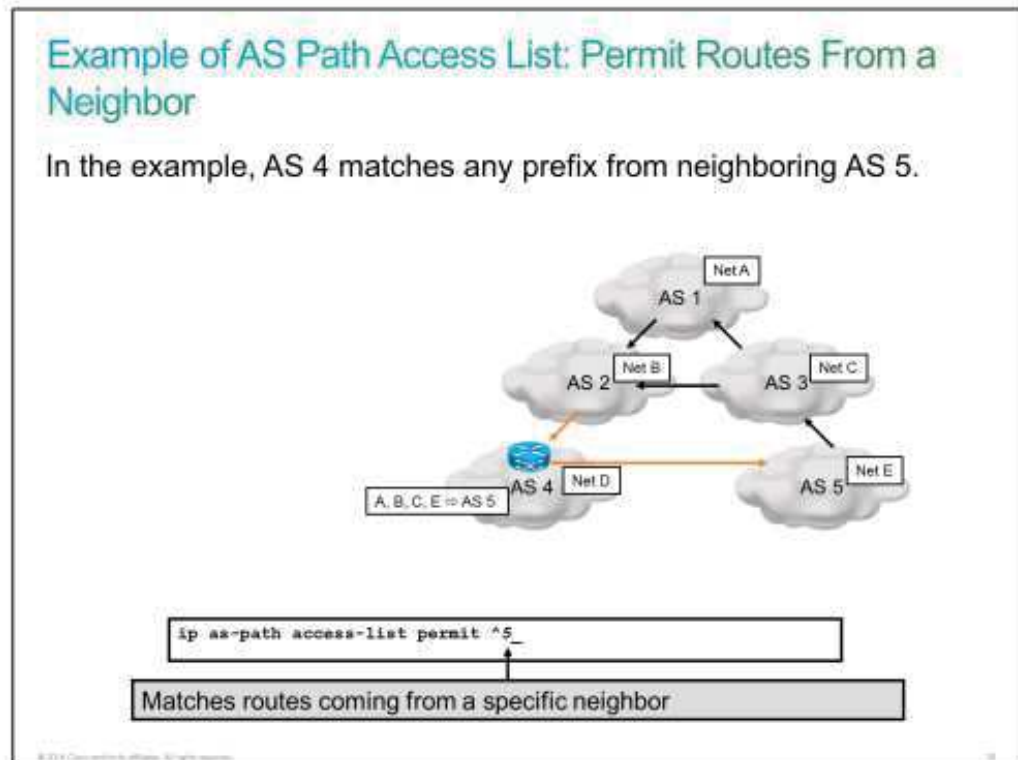


The sample regular expression (**ip as-path access-list permit ^\$**) matches any route that has an empty AS path attribute (that is, no character from start to end). Only locally originated routes have an empty AS path attribute, hence this regular expression is used when matching local routes.

This type of filter is used by multihomed customers to send only their address space to their service providers, to prevent them from becoming a transit AS. In the figure, AS 4 only advertises its own prefix ("D") to its providers. Other prefixes that are received from one provider are not forwarded to the other provider.

Example of AS Path Access List: Permit Routes From a Neighbor

The first number in the AS path is the last prepended number. The directly connected neighboring AS is always found as the first number in the AS path. A sample AS path access list to permit routes from a specific neighbor in Cisco IOS and IOS XE software is shown below.

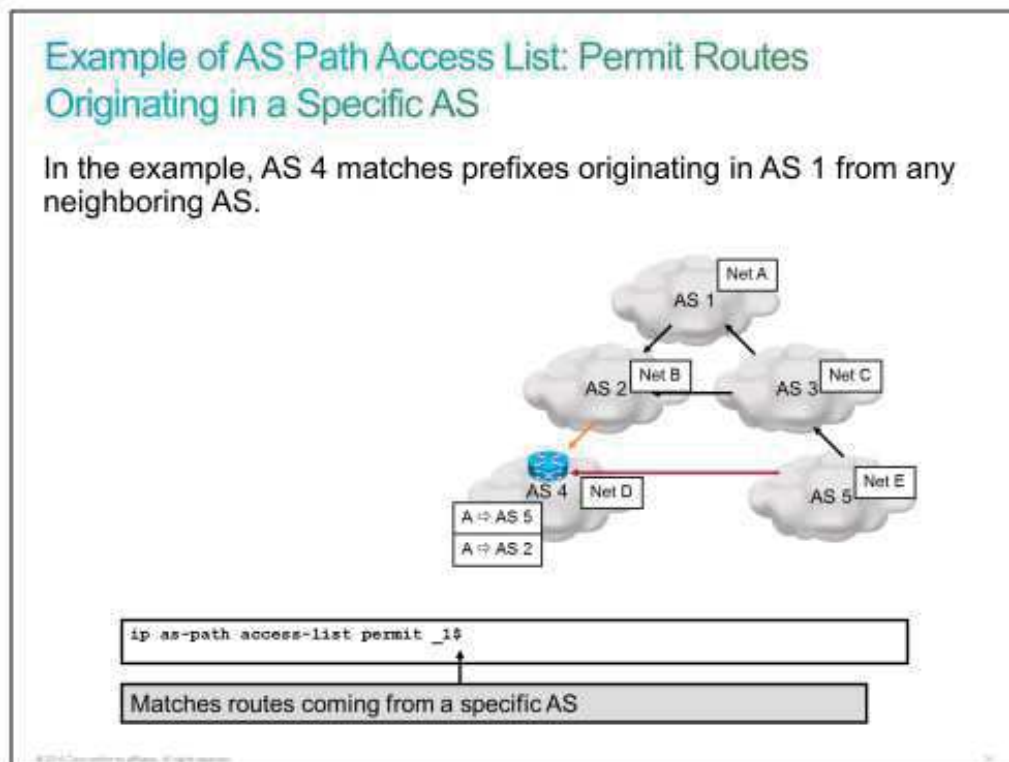


The figure illustrates a regular expression (`ip as-path access-list permit ^5_`) that matches any route received from a neighboring AS 5, the first number in the AS path. All prefixes received from AS 5 are accepted. If the same filter is applied to incoming updates from AS 2, the prefixes would be denied.

This type of filter is typically used when creating routing policies (for example, assigning different local preference values).

Example of AS Path Access List: Permit Routes Originating in a Specific AS

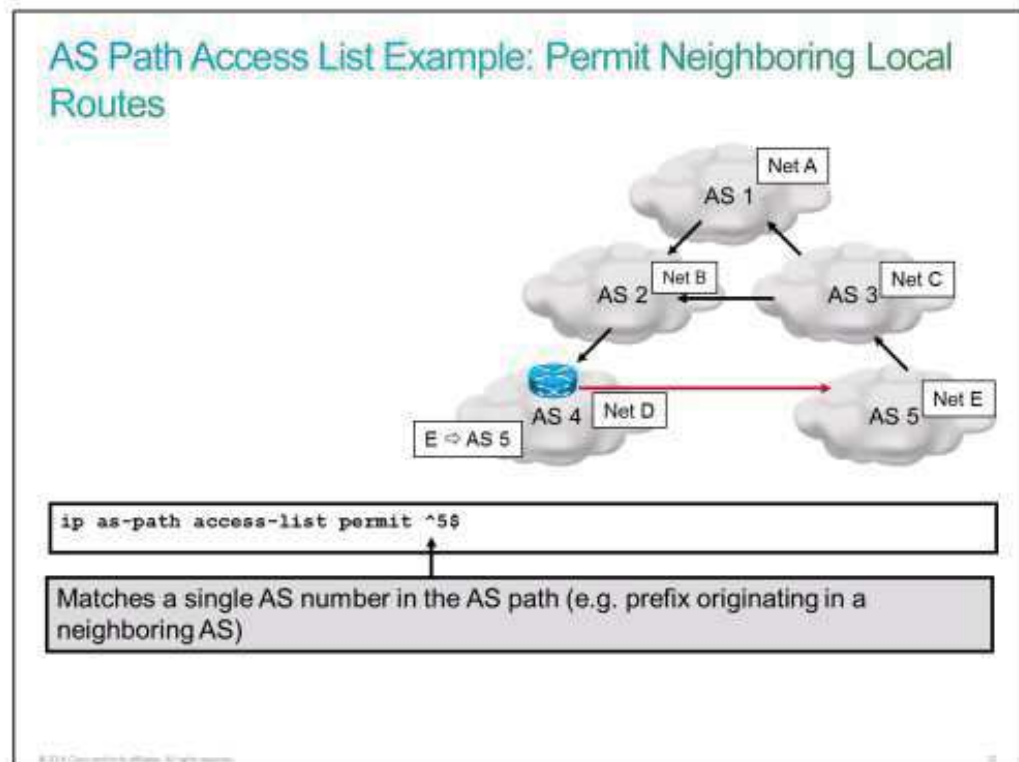
The last number in the AS path is the first prepended number. The originating AS is always found as the last number in the AS path. A sample AS path access list to permit routes from a specific AS in Cisco IOS and IOS XE software is shown below.



The figure illustrates a regular expression (`ip as-path access-list permit _1$`) that matches any route that is originated in AS 1, the last number in the AS path. If this filter is applied to incoming updates from AS 2 or AS 5, it will permit prefix "A" originating in AS 1. This type of filter is commonly used to implement routing policies where you can assign preference for certain prefixes coming from a preferred service provider.

Example of AS Path Access List: Permit Neighboring Local Routes

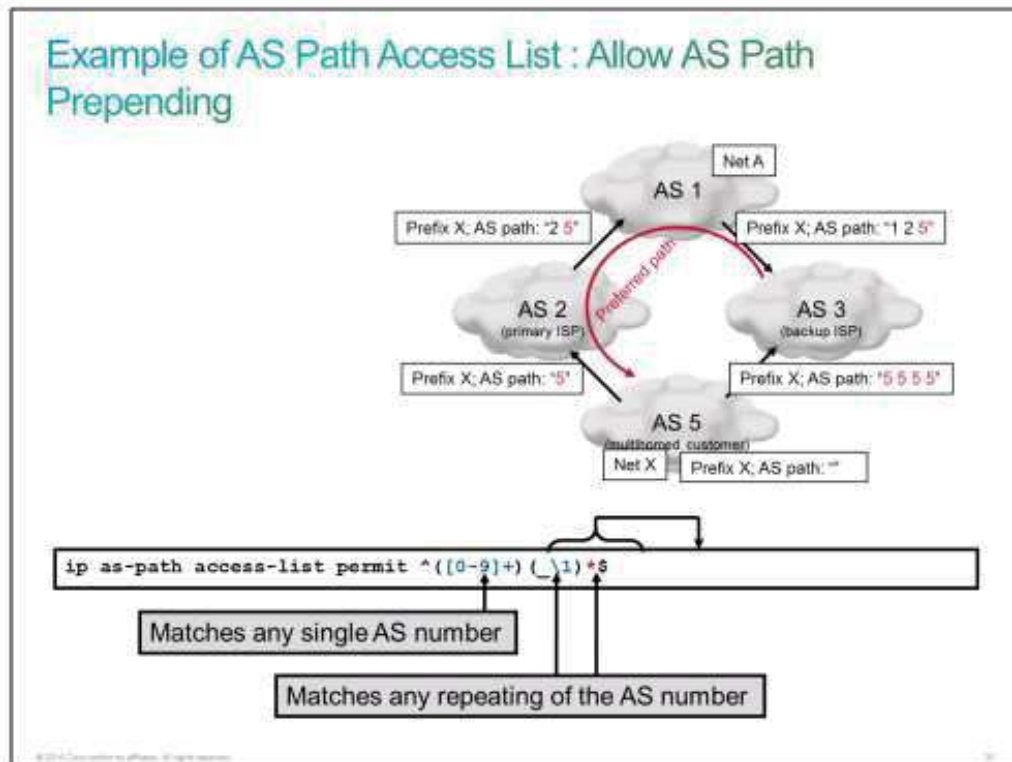
A single AS number in an AS path denotes prefixes originating in the neighboring AS. A sample AS path access list to permit routes that originate in a neighboring AS in Cisco IOS and IOS XE software is shown below.



The figure illustrates a regular expression (**ip as-path access-list permit ^5\$**) that matches any route that has originated in the neighboring AS 5, the first number in the AS path. In the example, AS 4 only accepts prefix “E” from AS 5, because other prefixes originate in other autonomous systems.

Example of AS Path Access List: Allow AS Path Prepending

The customer can signal a backup link using AS path prepending. Alternatively, a specific per-neighbor regular expression can be used (for example, “(5)(_5)*”). A sample AS path access list to permit local prefixes with AS path prepending in Cisco IOS and IOS XE software is shown here.



The figure illustrates a generic filter (**ip as-path access-list permit ^([0-9]+)(_1)*\$**) that can be used on any neighboring AS where you wish to accept the local prefixes of the neighbors, while still allowing them to perform AS path prepending.

Enclosed within the first parentheses is a range of digits that can appear multiple times. In BGP, this effectively matches any number from 0 to 65535. Enclosed within the second pair of parentheses, you reference whatever was matched in the first parentheses, and allow that number to repeat zero or more times.

AS 3 can reach prefix X via AS 1 or via AS 5 directly. However, since AS 5 is using AS path prepending for its updates to AS 3, AS 3 will prefer the seemingly shorter AS path that is through AS 1 to reach prefix X.

Summary

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

Summary

- OSPF routing updates can be filtered based on prefix and prefix length, LSA type, and route source.
- BGP routing updates can be filtered based on prefix and prefix length, next-hop address, route source, AS path, BGP communities, and local preference.
- You can use the following route filtering tools: prefix lists, AS path access lists, route maps, and route policy language.
- The typical filtering objective in OSPF is to filter redistributed routes on the ASBR.
- The typical filtering objective in IS-IS is to filter redistributed routes.
- The typical filtering objective in BGP is to implement inbound or outbound filtering based on customer requirements.
- Prefix lists are designed to simplify the filtering of routing updates.
- Regular expressions are used to identify AS numbers in AS path access lists.

© 2014 Cisco and/or its affiliates. All rights reserved.

39

Lesson 2: Introducing Route Maps and Routing Policy Language

Overview

The lesson focuses on two mechanisms that are available, in combination with routing protocols, to filter routing information or to implement desired routing policies using Cisco IOS or IOS XE and Cisco IOS XR routers.

The lesson describes route maps that are available on Cisco IOS and IOS XE routers, and the RPL, that is available on Cisco IOS XR routers. The lesson also describes how to manipulate BGP attributes using RPL and how to use value sets.

Upon completing this lesson, you will be able to:

- Explain an overview of route maps
- Describe route maps processing when processing a routing update
- Describe route maps syntax
- Explain a route map example
- Describe the characteristics of RPL
- Explain an RPL example
- Describe RPL pass and drop actions
- Describe RPL conditions
- Describe RPL operators
- Describe RPL Boolean operators
- Describe how to nest statements in RPL

- Describe how to set attributes and parameters in RPL
- Describe how to set BGP attributes and parameters using RPL
- Show an example of setting BGP attributes and parameters using RPL
- Describe how to set OSPF and IS-IS parameters using RPL
- Describe how to use parameterization in RPL
- Describe how to apply routing policies
- Describe how to maintain routing policies
- Describe value sets that can be used in RPL
- Describe AS path sets
- Describe standard community sets
- Describe prefix sets
- Describe how to monitor routing policies
- Describe how to test routing policies
- Describe how to translate route maps to routing policies

Route Maps Overview

Route maps are a simple language to support complex routing policies, in addition to filtering.

Route Maps Overview

Route map characteristics:

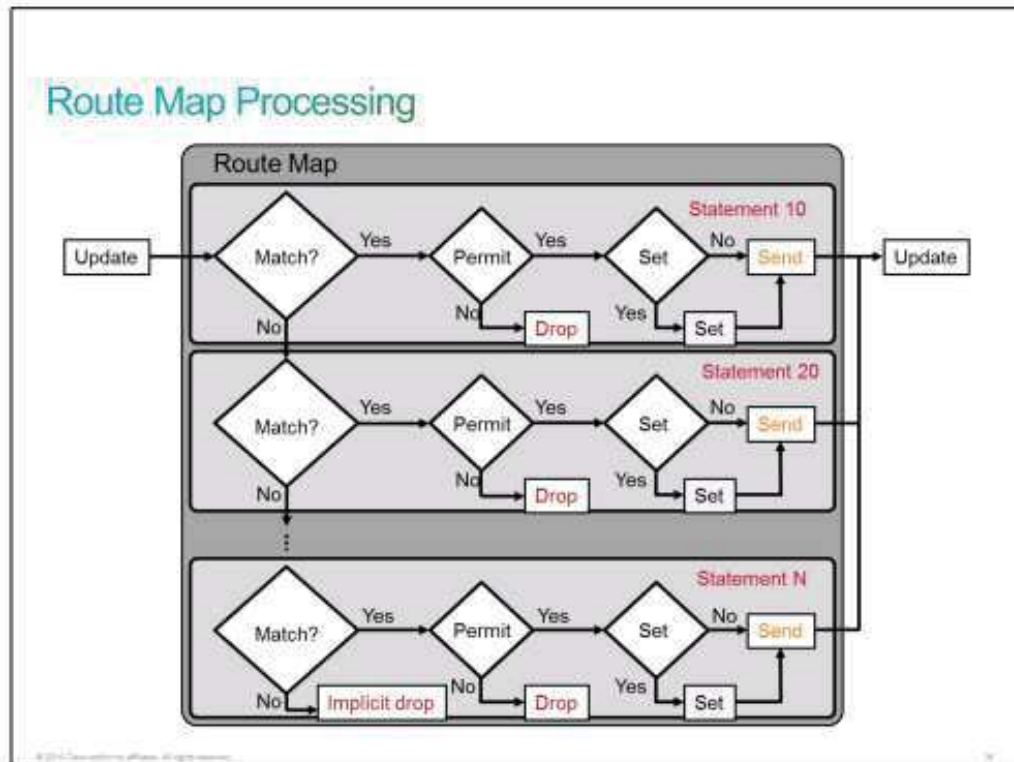
- Route maps are a simple language to support complex routing policies, in addition to filtering.
- Route maps are uniquely identified by a case-sensitive name.
- Each route map consists of one or more statements.
- Each statement contains zero or more **match** commands.
- Each statement contains zero or more **set** commands used to modify routing updates.
- Route maps are available in Cisco IOS or IOS XE Software. (Cisco IOS XR Software uses the RPL)

Route maps are uniquely identified by a case-sensitive name and consist of one or more statements. Each statement contains zero or more **match** and **set** commands. The **match** command is used to identify which routes should be processed in a given statement. The **set** command specifies which parameters should be modified or added in a routing update.

Route maps are not available in Cisco IOS XR Software. Instead, Cisco IOS XR Software uses the RPL.

Route Map Processing

Route maps process a route in the order defined by sequence numbers.



The diagram illustrates the components and processing of a route map:

- A route is processed by route map statements in the order defined by sequence numbers.
- If a route matches the match conditions, it is processed by that statement.
- If the statement uses the “**deny**” options, the route is immediately dropped.
- If a route has one or more **set** commands, the **set** commands are processed, resulting in modified or added parameters and attributes.

Route Maps Syntax

In addition, route maps also have the **continue** command, allowing the processing to continue in another statement (that is, the jump command).

Route Maps Syntax

Additional route-map options:

- The **continue** command can be used to jump to another statement instead of exiting.
- Policy lists can be used to modularize and group **match** statements.

```
route-map Policy1 permit 10
match condition
continue 40
!
route-map Policy1 permit 20
...
!
route-map Policy1 permit 40
...
!
route-map Policy1 permit 1000
```

```
route-map Policy2 permit 10
match policy-list Policy3
```

```
route-map Policy3 permit 10
...
route-map Policy3 permit 20
...
route-map Policy3 permit 30
```

Complex match options can be grouped in policy lists and then reused in various route maps for more modularity and reusability.

Route Maps Syntax (Cont.)

Route maps syntax characteristics:

- Each route map is identified using a case-sensitive name.
- Each route map can have one or more ordered statements identified using the sequence number.
- Each route-map statement can filter updates using **permit** or **deny** options.
- Each statement processes updates matched by the **match** command.
- Each statement can optionally modify or set parameters in an update.
- Match conditions of the same type are evaluated using a logical **OR** operator; match conditions of different types are evaluated using a logical **AND** operator.

```
Router(config)# route-map map-tag [permit | deny] [sequence-number]
match condition
match condition
set parameter value
set parameter value
```

A route map statement processes routes that are matched by a match condition (for example, any prefix permitted by a prefix list will be matched; any prefix denied by a prefix list will not be processed by the statement, and is instead evaluated by the next route map statement). If a route is matched, it can then be permitted or denied by the route map statement.

If there are multiple match conditions, they are evaluated using the following rules:

- Match conditions of the same type are evaluated using the logical OR operator (that is, the prefix must be matched by at least one condition).
- Match conditions of different types are evaluated using the logical AND operator (that is, the prefix must be matched by all conditions).

Routes matched and permitted by a statement can optionally be modified using **set** commands.

Example of Route Maps

This topic provides a route map example.

Example of Route Maps

Route maps examples:

- Preferred paths for specific prefixes
- Backup paths for specific prefixes
- Preferred paths for prefixes based on AS path
- Backup paths for prefixes based on AS path
- Explicit permit at the end

```
route-map Policy1 permit 10
  match ip address prefix-list PL1
  set local-preference 200
!
route-map Policy1 permit 20
  match ip address prefix-list PL2
  set local-preference 50
!
route-map Policy1 permit 30
  match as-path APACL1
  set local-preference 200
!
route-map Policy1 permit 40
  match as-path APACL2
  set local-preference 50
!
route-map Policy1 permit 1000
```

The sample route map configuration consists of five route map statements. The first two process routes are matched by a prefix list based on the prefix and set appropriate BGP local preference attributes. The next two statements match routes using AS path access lists and also set appropriate BGP local preference values. All nonmatched routes are passed unchanged by explicitly permitting them at the end.

Example of Route Maps (Cont.)

Additional route map example:

- The first route-map statement processes routes matched by prefix list PL1 or PL2 and AS path access list APACL1.
- These routes are assigned local preference 100 and MED 1000.
- All other routes are passed unchanged.

```
route-map Policy1 permit 10
 match ip address prefix-list PL1 PL2
 match as-path APACL1
 set local-preference 200
 set metric 1000
!
route-map Policy1 permit 1000
```

© 2014 Cisco Systems, Inc. All rights reserved.

6

The sample configuration illustrates the logical processing of multiple match conditions. The first condition uses two prefix lists where a route must be matched by at least one of the prefix lists (logical OR). In order for a route to be processed by this statement, it must also match the second match command, which uses an AS path access list (logical AND).

A single match statement may contain multiple conditions of the same type (prefix lists PL1 and PL2 in this example). At least one condition in the match statement must be true for that match statement to be considered a match (logical OR).

A route map statement may also contain multiple match statements of different types (prefix lists and AS path access lists in this example). All match statements must be true for the route map statement to be considered a match (logical AND).

This example can be illustrated as “(PL1 **OR** PL2) **AND** APACL1”.

Routing Policy Language

The RPL is a newer mechanism that was introduced into Cisco IOS XR Software as a replacement and improvement upon the route maps used in Cisco IOS or IOS XE Software.

Routing Policy Language

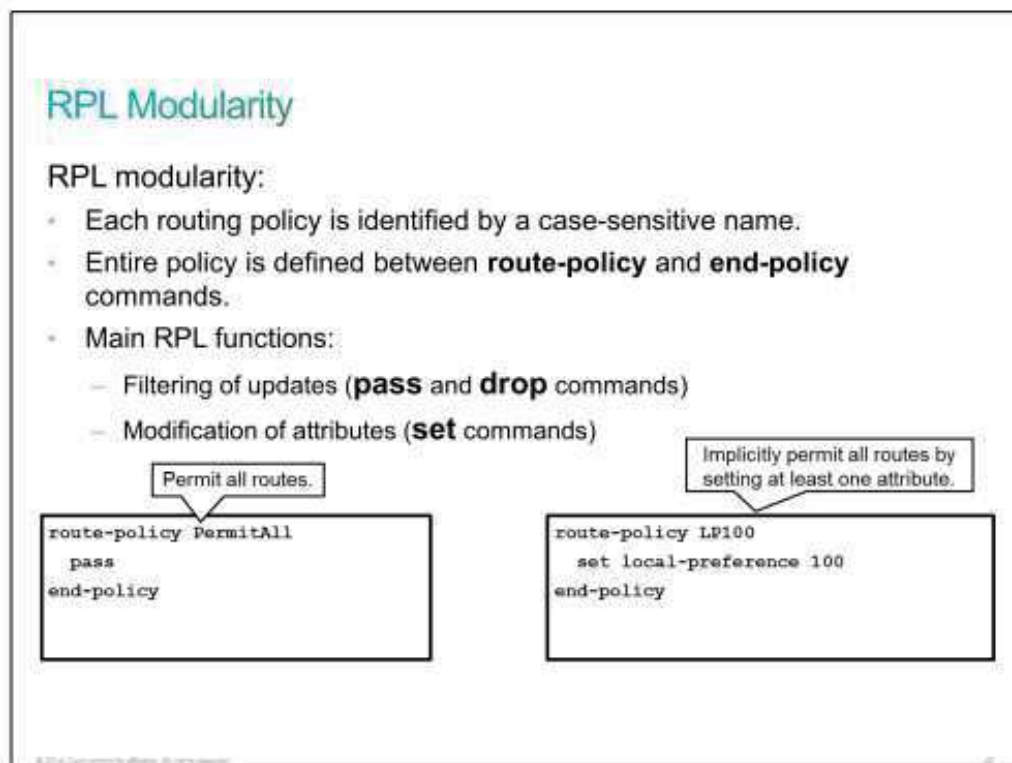
RPL characteristics:

- RPL replaces route maps in Cisco IOS XR Software.
- RPL is a simple, yet powerful language, designed to process routing updates.
- RPL addresses the deficiencies of route maps in Cisco IOS or IOS XE Software:
 - Better modularity
 - Better reusability
 - Parameterization
 - Nesting of policies and conditions
 - Powerful match options
 - Reusable value sets

RPL offers a more powerful set of tools to process routes:

- Modularity allows policies to reference other objects such as prefix list, value sets, and other policies (that is, nesting of policies).
- Parameterization permits optimization and better reusability of policies.

RPL Modularity



Like route maps or many other objects, routing policies are identified by using a case-sensitive name. Each routing policy is a single object (no sequence numbers or multiple lines or statements).

Like route maps, routing policies are also a filtering tool that allows you to permit or deny routing updates. The explicit commands to permit or deny are **pass** and **drop** respectively.

Like route maps, routing policies can also modify or add parameters or attributes using the **set** command. A single **set** command also implicitly includes the **pass** command.

RPL Example

This sample configuration shows how to enable the forwarding of routing updates to an external BGP neighbor.

RPL Example

EBGP:

- Note: Cisco IOS XR Software does not automatically send BGP updates to external peers.
- A routing policy is required to forward updates.

Permit all routes to EBGp peers.

```
route-policy PermitAll
  pass
end-policy
!
router bgp 1
  neighbor 1.2.3.4
    remote-as 64111
    address-family ipv4 unicast
      route-policy PermitAll out
    !
  !
  !
```

In Cisco IOS XR Software, updates are not forwarded to an external neighbor unless an outbound policy is attached to the neighbor.

The sample configuration uses a simple policy to permit all routes.

RPL Pass and Drop Actions

The sample configurations illustrate the routing policy pass and drop actions.

RPL Pass and Drop Actions

Routing policy rules:

- Using the explicit **pass** command continues the processing of route policy.
- Using the explicit **drop** command stops processing of route policy.
- The default action is **drop**.
- If any modification is applied to a route (e.g. **set**), it is an implicit **pass**.

```
route-policy DropOrPass1
end-policy

route-policy DropOrPass2
pass
end-policy

route-policy DropOrPass3
drop
end-policy

route-policy DropOrPass4
set med 100
end-policy

route-policy DropOrPass5
pass
drop
pass
end-policy
```

Drop!

Pass!

Drop!

Pass!

Drop!

Routing policy rules are listed here:

- An empty policy implicitly denies all routes.
- An explicit **pass** without any conditions will forward all routes without any modifications.
- An explicit **drop** will do the same as an implicit **drop** (deny all routes).
- A **set** command will modify the attribute accordingly and forward all routes.
- An explicit **drop** command will stop the processing of a policy and deny a route.

RPL Conditions

RPL uses various match options for conditional update processing.

RPL Conditions

```
if attribute operator value then
  ... do something ...
elseif attr operator value then
  ... do something else ...
else
  ... do something else ...
endif
```

```
route-policy SetLP
  if med eq 10 then
    set local-preference 200
  elseif med eq 20 then
    set local-preference 150
  else
    set local-preference 50
  endif
end-policy
```

© 2014 Cisco and/or its affiliates. All rights reserved.

53

RPL uses the conditional statement syntax that is found in many programming languages:

```
if condition then operation1 else operation2 endif
```

```
if condition1 then operation1 elseif condition2 then operation2 else operation3 endif
```

The sample configuration illustrates how the **MED** attribute can be used to influence a routing policy by setting a more powerful, local preference attribute.

RPL Operators

Comparing attributes against values supports different operators.

The diagram consists of a large rectangular frame. In the top-left corner of the frame, the text "RPL Operators" is written in a teal color. In the center-right of the frame, there is a smaller rectangular box with a black border containing a code snippet. To the left of this code box, there is a callout box with a black border and a tail pointing towards the code box. The callout box contains the text "Simple conditions".

RPL Operators

Simple conditions

```
route-policy SetLP
  if med le 19 then
    set local-preference 200
  elseif med eg 20 then
    set local-preference 150
  elseif med ge 21 then
    set local-preference 50
  endif
end-policy
```

RPL conditions can use a variety of operators:

- **eq** : An attribute numerically equal to a specified value
- **le** : An attribute numerically lower than or equal to a specified value
- **ge** : An attribute numerically greater than or equal to a specified value
- **is** : An attribute equal to a specified value (used for non-numerical values)
- **in** : An attribute contained in a value set
- Many attribute-specific conditions (for example, AS path matching)

RPL Boolean Operators

Boolean operators can be used to create complex compound conditions.

RPL Boolean Operators

Multiple match options can be combined using Boolean operators:

- **and** : both conditions must match
- **or** : at least one condition must match
- **not** : negate the following condition

Using composite conditions

```
route-policy SetLP
  if med eq 10 and not local-preference eq 100 then
    set local-preference 200
  elseif med eq 20 or local-preference eq 200 then
    set local-preference 150
  else
    set local-preference 150
  endif
end-policy
```

© 2014 Cisco and/or its affiliates. All rights reserved.

55

Boolean operators can be used as listed here:

- Use the **and** operator if two or more conditions must match.
- Use the **or** operator if at least one of two or more conditions must match.
- Use the **not** operator to negate a condition.

RPL Boolean Operators (Cont.)

Multiple match options can be combined using Boolean operators:

- **not** : highest precedence
- **and** : higher precedence than or, lower than not
- **or** : lowest precedence
- Influence precedence by grouping using parentheses.

```
if med eq 10 and not local-preference eq 100 or med eq 50 then
```

versus

```
if med eq 10 and (not local-preference eq 100 or med eq 50) then
```

versus

```
if med eq 10 and not (local-preference eq 100 or med eq 50) then
```

© 2014 Cisco Systems, Inc. All rights reserved.

45

Use parentheses to influence the precedence of operators and achieve the desired result. The operators have the following precedence:

- **not** is always evaluated first.
- **and** is evaluated second.
- **or** is evaluated last.

The first example is not using any parentheses. It can be written with parentheses to ensure the proper understanding of the condition:

```
if ((med eq 10) and (not (local-preference eq 100))) or (med eq 50) then
```

The second and third example will result in different conditions.

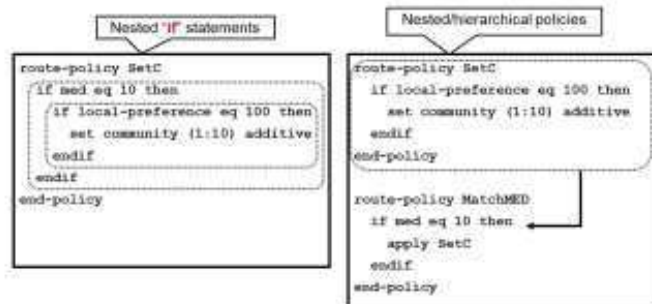
RPL Nesting

Large and complex routing policies should preferably be optimized by using modularization as much as possible.

RPL Nesting

RPL nesting characteristics:

- Two types of nesting are supported:
 - “if” statement within another “if” statement
 - A routing policy within another routing policy
- Multiple levels of nesting are supported.

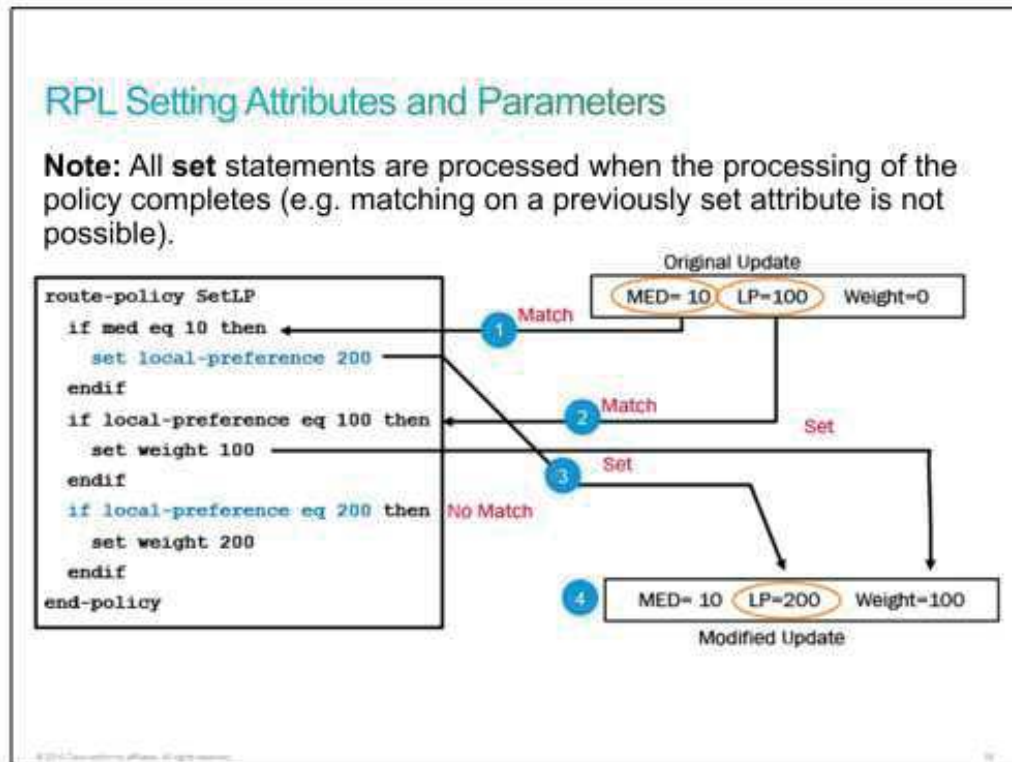


A nested if statement is shown in the left-side example and a nested, also called hierarchical, route-policy is shown in the right-side example.

The **SetC** policy in the right-side example can be reused in multiple policies to conditionally assign a [BGP](#) community. The **apply** command is used within a route-policy to call another route-policy.

RPL Setting Attributes and Parameters

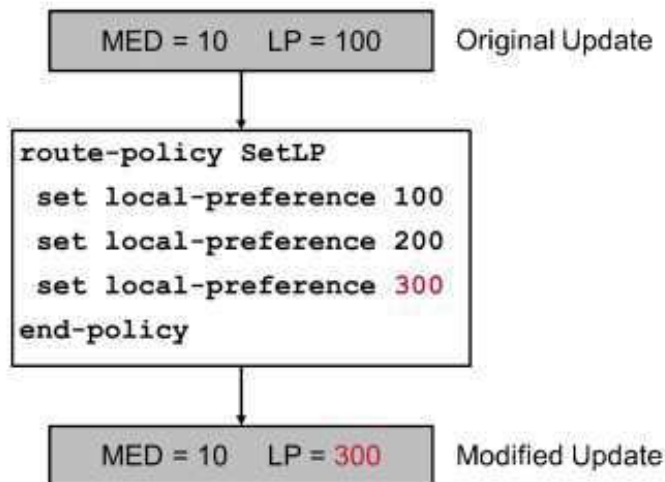
Use the **set** command to assign values to attributes and parameters.



The figure illustrates a policy with multiple conditions and modifications that are based on the same parameter (BGP local preference). The important point to remember is that the modifications of an attribute are only executed when the policy completes, and that conditions that are based on a previously modified value cannot be used.

RPL Setting Attributes and Parameters (Cont.)

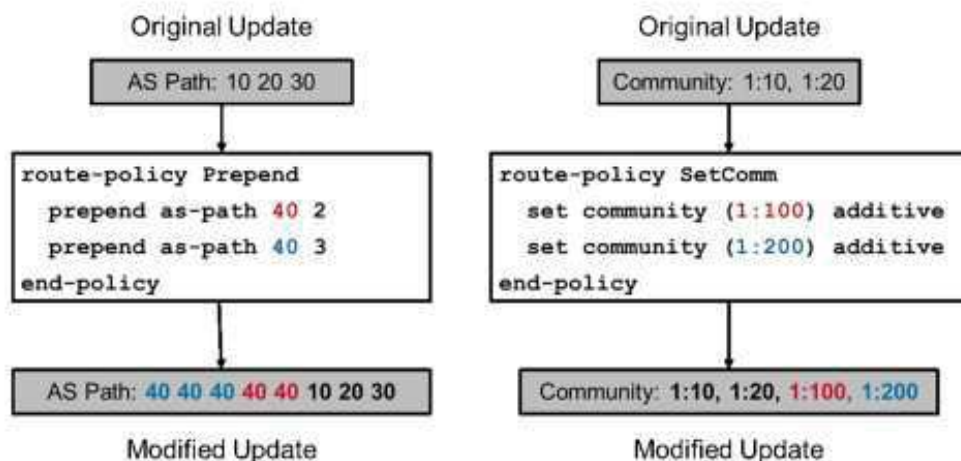
Note: Last **set** wins when multiple sets are evaluated for a unique parameter.



If multiple **set** commands are processed for the same attribute, the last one will be used when the processing of the policy completes.

RPL Setting Attributes and Parameters (Cont.)

Note: All **set** commands are evaluated in the same order for nonunique attributes and operations.



If a **set** command is processing an attribute that is a set of values, it may happen that all **set** commands will have effect.

The left-side example shows how the first **prepend** command modifies the AS path attribute by prepending 40 twice. The second **prepend** command then additionally prepends 40 three times.

The right-side example shows how two **set** commands add two values to a set of BGP community attributes.

RPL Setting BGP Attributes and Parameters

RPL Setting BGP Attributes and Parameters

BGP attributes and parameters, RPL characteristics:

- Standard BGP community attribute:

```
set community {value [value2 ...]} [additive]
```

- Extended BGP community attribute:

```
set extcommunity {value [value2 ...]} [additive]
```

© 2014 Cisco and/or its affiliates. All rights reserved.

59

RPL Setting BGP Attributes and Parameters (Cont.)

- BGP dampening parameters:

```
set dampening [half-life value] [max-suppress value] [reuse value] [suppress value]
```

- Local preference attribute:

```
set local-preference value
```

- MED attribute:

```
set med {[+|-]value | igp-cost | max-reachable}
```

© 2014 Cisco and/or its affiliates. All rights reserved.

60

Setting standard and extended BGP community attributes:

- One or more values can be assigned to the BGP community attribute.
- If the **additive** keyword is used, the new communities will be added to the existing BGP communities.
- Omitting the **additive** keyword will result in the existing BGP communities being overwritten.

BGP route-flap dampening can be enabled and tuned using the **set dampening** command, where the dampening parameters can be specified to determine the aggressiveness of the dampening mechanism.

BGP local preference can be set on incoming updates or upon redistribution. The default BGP local preference is 100 and is set on all updates coming from external peers or being redistributed.

The BGP MED attribute can be set to a specific value or modified using the "+" or "-" options. MED can also be set to the underlying IGP cost or to a maximum value.

RPL Setting BGP Attributes and Parameters (Cont.)

BGP attributes and parameters RPL characteristics:

- Delete standard BGP community attributes:

```
delete community {all | [not] in community-set}
```

- Delete extended BGP community attributes:

```
delete extcommunity rt {all | [not] in extcomm-set}
```

- Prepend AS path:

```
prepend as-path {AS | most-recent} [count]
```

© 2014 Cisco and/or its affiliates. All rights reserved.

10

RPL Setting BGP Attributes and Parameters (Cont.)

- Replace a sequence of AS numbers with local AS:

```
replace as-path {private-as | 'AS1 AS2 ...'}
```

- Suppress route if aggregated:

```
suppress-route
```

- Unsuppress route if aggregated:

```
unsuppress-route
```

© 2014 Cisco and/or its affiliates. All rights reserved.

19

The **delete** command can be used in combination with standard and extended BGP communities to delete some or all of the BGP community attributes.

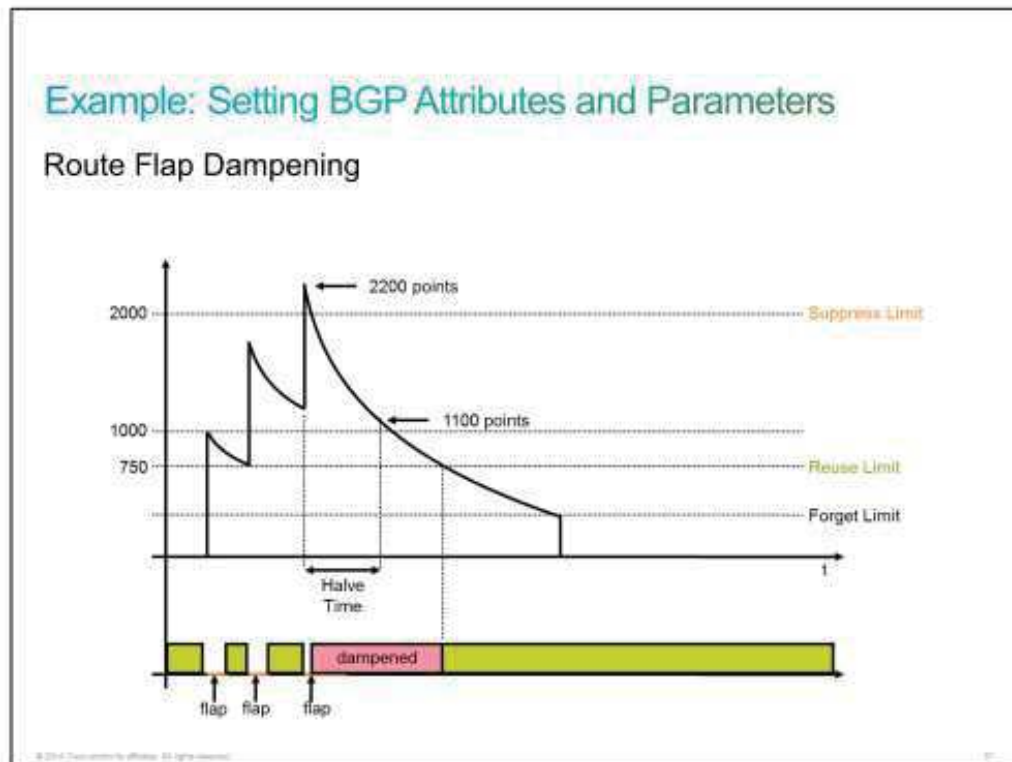
The **prepend as-path** command can be used to prepend an arbitrary number to the AS path a number of times.

The **replace as-path** command can be used to replace all occurrences of private AS numbers with the local AS number, or to arbitrarily replace specified AS numbers with the local AS number.

Policies can be used in combination with summarization (aggregation) in order to set various parameters to the summary, but also to specify which individual routes are suppressed or unsuppressed.

Example: Setting BGP Attributes and Parameters

BGP route-flap dampening is a feature designed to make BGP more stable and consequently scale better by “punishing” routes that flap (disappear and reappear) more often. An example of setting BGP attributes and parameters in case of route-flap dampening is shown below.



The default behavior of dampening results in stopping propagation of routes that consecutively flap three or more times in a short period, for a certain period of time.

The default behavior can be summarized:

- Each flap is penalized by adding 1000 penalty points to the penalty.
- If a cumulative penalty exceeds the **suppress limit** (2000 points by default), the route is dampened (for example, it is stored in the BGP table, but is not evaluated in the best-path selection and is consequently not installed into the routing table nor forwarded to any neighbor). The penalty is remembered by routers when the route is not reachable by storing it as a “history” entry.
- The penalty is gradually decreased. The penalty reduction is determined by the **half-time**, which is 15 minutes by default.
- When a penalty drops below the **reuse limit** (750 by default) or when the route has been dampened for more than the **maximum suppress time** (one hour by default), the route becomes valid again.
- When the penalty drops below one half of the **reuse limit**, all flap history and penalty is forgotten.

Example: Setting BGP Attributes and Parameters (Cont.)

Conditional BGP dampening

Conditional BGP dampening, where smaller prefixes are more aggressively punished than larger prefixes.

```
router bgp 1
  address-family ipv4 unicast
    bgp dampening route-policy BDamp
  !
route-policy BDamp
  if destination in (0.0.0.0/0 ge 25) then
    set dampening max-suppress 30 halflife 10 reuse 750 suppress 1000
  elseif destination in (0.0.0.0/0 ge 21) then
    set dampening max-suppress 15 halflife 7 reuse 750 suppress 2000
  elseif destination in (0.0.0.0/0 ge 17) then
    set dampening max-suppress 10 halflife 5 reuse 750 suppress 3000
  else
    set dampening max-suppress 5 halflife 3 reuse 750 suppress 4000
  endif
end-policy
```

The sample configuration illustrates how graded BGP route-flap dampening is configured:

- Small prefixes (/25 to /32) are assumed to be more likely to flap, and are hence more aggressively punished if they flap several times.
- Larger prefixes (/21 to /24) are assumed to be slightly more stable and are less aggressively punished (allow more flaps before suppression and become unsuppressed faster when they stabilize).
- Large prefixes (/17 to /20) are even less aggressively punished if they flap. The largest prefixes (/0 to /16) are assumed to be the most stable (large summaries belonging to service providers) and are suppressed after more than four consecutive flaps and are unsuppressed within ten minutes after stabilizing.

RPL Setting OSPF and IS-IS Parameters

Routing policies can also be used in combination with OSPF and IS-IS to modify the routing information.

RPL Setting OSPF and IS-IS Parameters

The following OSPF and IS-IS parameters can be changed:

- OSPF metric type:

```
set metric-type {type-1 | type-2}
```

- OSPF metric:

```
set ospf-metric value
```

© 2014 Cisco Systems, Inc. All rights reserved.

18

RPL Setting OSPF and IS-IS Parameters (Cont.)

- IS-IS metric type:

```
set metric-type {external | internal}
```

- IS-IS metric:

```
set isis-metric value
```

- IS-IS level for redistributed routes:

```
set level {level-1 | level-2 | level-1-2}
```

© 2014 Cisco Systems, Inc. All rights reserved.

19

OSPF and IS-IS parameters such as metric, metric type, and level for redistributed routes can be changed. To change the metric type, use the **set metric-type {type-1 | type-2}** command with OSPF and the **set metric-type {external | internal}** command with IS-IS. Note that IS-IS and OSPF do not have the same metric types, so you cannot use a unique command.

You can set the metric using the **set ospf-metric value** command with OSPF that is similar to IS-IS, where you can set the metric using the **set isis-metric value**.

To change which routes should be redistributed in IS-IS, use the **set level {level-1 | level-2 | level-1-2}** command.

RPL Parameterization

In order to make policies modular and reusable, parameters can be used in place of fixed values when calling nested policies.

RPL Parameterization

RPL supports two types of parameters:

- **Global parameters:**
 - Defined globally using the **policy-global** command
 - Available for use in all routing policies
- **Parameters passed to a nested routing policy:**
 - Defined when creating a routing policy
 - Available in **match** and **set** statements within a policy or when calling another nested routing policy

A policy can reference global parameters or parameters passed to it from a calling policy.

RPL Global Parameters

RPL Global Parameters

RPL global parameters characteristics:

- Parameters are defined using the **policy-global** command, and are separated by commas.
- Values are defined within single quotes.
- Parameters are referenced by prepending the **\$** sign to the name of the parameter.

Defining global variables

```
policy-global
# Global variables
AS '65001',
Lo0 '10.1.2.3',
EBGP1 '192.168.1.1',
EBGP2 '192.168.2.1',
DefWeight '0',
DefLP '100',
DefMED '0'
end-global
```

Using global variables

```
route-policy SetMED
if as-path originates-from '$AS' then
  set med $DefMED
endif
end-policy
```

The left-side example illustrates the usage of the **policy-global** commands where all the global variables should be defined. These variables can then be referenced by any routing policy.

The right-side example illustrates a routing policy referencing two global variables.

RPL Passed Parameters

RPL Passed Parameters

RPL passed parameters characteristics:

- Declare parameters when creating a routing policy.
- Nesting policies with parameters allows for greater modularization and optimization of policies.

Policy using passed parameters

```
route-policy SetMED($med, $as)
if as-path originates-from '$as' then
  set med $med
else
  set med max-reachable
endif
end-policy
```

Using a nested policy and passing parameters to it

```
route-policy ProcessUpdates
if as-path neighbor-is '100' then
  apply SetMED(50,100)
elseif as-path neighbor-is '200' then
  apply SetMED(150,200)
endif
end-policy
```

The sample configuration illustrates a modular approach to creating routing policies, where one policy (on the right) calls another policy (on the left) that applies a different MED value, based on whether the route originates in a given AS. Note that matching based on the AS path is always done using regular expressions, which must be enclosed within single quotes.

The left-side routing policy is defined with two parameters: **\$med** and **\$as**. When you are calling this policy from within another policy, using the `apply` command, you should supply two parameters.

Applying Routing Policies

When building a routing policy, clearly defining the requirements for the route-policy is important.

Applying Routing Policies

Steps to apply routing policies:

- Design a routing policy.
- Configure the policy.
- Test the policy by using **show** commands.
- Apply the policy if it is correct.

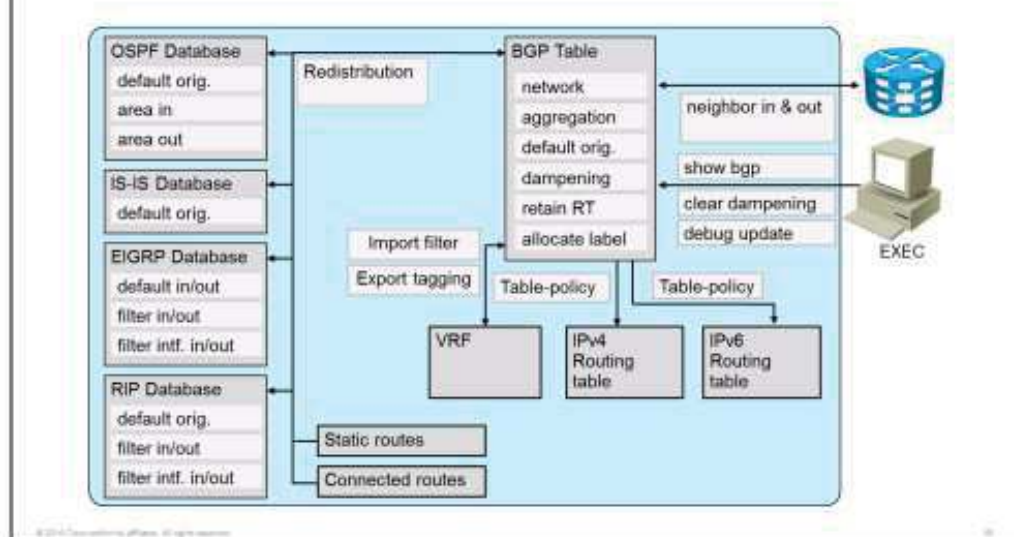
Use routing policies in many places (attach points):

- Routing updates (e.g. BGP, OSPF, EIGRP, IS-IS, RIP)
- Route origination (e.g. redistribution, network commands)
- Route insertion into routing table
- **show** commands to filter output

Often a route-policy will be derived from existing routers using Cisco IOS or IOS XE Software, in which case it requires a route map to be “translated” to [RPL](#). Reviewing a route-policy and optimizing it is recommended in order to simplify its maintenance.

Applying Routing Policies (Cont.)

Attach Points



The figure illustrates the many attach points for routing policies:

- Redistribution between any pair of routing protocols
- Received or sent updates, depending on the limitations of routing protocols (for example, [ABRs in OSPF](#))
- Origination of routes in BGP by using network statements or summarization
- Injecting routes into the routing table from BGP
- Using **show** commands in BGP to filter the output or test the effect of the routing policy

Applying Routing Policies (Cont.)

Validity Checking:

RPL validity checking is done in two phases:

1. Syntax checking and value checking are performed during policy configuration.

```
RP/Q/RP1/CPU0:CRS(config-rpl)# set med 289314790283408912634789
```

```
% Invalid input detected at '^' marker.
```

© 2014 Cisco and/or its affiliates. All rights reserved.

72

Applying Routing Policies (Cont.)

2. Applicability of a policy for a given attach point is checked during configuration commit.

```
RP/Q/RP1/CPU0:CRS(config-bgp-af)# commit
```

```
% Failed to commit one or more configuration items during an atomic operation,  
no changes have been made. Please use 'show configuration failed' to view the  
errors
```

```
RP/Q/RP1/CPU0:CRS(config)# show config failed
```

```
!! CONFIGURATION FAILED DUE TO SEMANTIC ERRORS
```

```
router bgp 1
```

```
address-family ipv4 unicast:
```

```
redistribute connected route-policy t9
```

```
!!! Could not find entry in list: Policy [t9] uses the 'ospf-metric' attribute.  
There is no 'ospf-metric' attribute at the BGP redistribution-dflt attach point.
```

© 2014 Cisco and/or its affiliates. All rights reserved.

73

Cisco IOS XR Software performs validity checks in two phases:

- Basic syntax and value checking are performed when a command is entered. The first example illustrates the configuration of MED with a value that is out of range. The syntax checking will reject this command immediately.

- The applicability of a routing policy is verified for a given attach point when the configuration is committed. The second example illustrates that the configuration of a route policy was successful, and the policy was successfully applied to redistribution into BGP. However, when you try to commit the configuration, the router will reject the configuration because there was an attempt to set OSPF parameters inside BGP.

Maintaining Routing Policies

Editing a routing policy requires the use of one of two available editors. Using the configuration mode approach will result in the policy being rewritten.

Maintaining Routing Policies

Routing policies maintenance:

- Trying to edit an existing routing policy through configuration-mode CLI will result in the policy being rewritten:

```
RP/0/RP1/CPU0:CRS(config)# route-policy R1
% WARNING: Policy object 'route-policy R1' exists! Reconfiguring it via CLI
will replace current definition. Use 'abort' to cancel.
RP/0/RP1/CPU0:CRS(config-rpl)# abort
```

- Use EXEC-mode editor instead.
- In Cisco IOS XR, three editors are accessible through EXEC mode:
 - GNU Nano (the default editor since Cisco IOS XR Release 3.6)
 - Micro Emacs
 - VIM
- Modify the policy, and complete the process:
 - Save changes, exit editor, commit changes

Upon exiting from the editor, you will be prompted to save and commit the changes.

The example shows the warning that is displayed when you try to go into policy configuration mode for an already configured policy. If this happens, the entire policy will be overwritten by the new configuration.

Maintaining Routing Policies (Cont.)

Using an Editor:

- An editor can be used for routing policies and sets.

```
RP/0/RP1/CPU0:CRS# edit ?
  as-path-set      edit an as-path-set
  community-set    edit a community-set
  extcommunity-set edit an extended-community-set
  policy-global    edit policy-global definitions
  prefix-set       edit a prefix-set
  rd-set           edit a rd-set
  route-policy     edit a route-policy
```

- Invoke the desired editor.

```
RP/0/RP1/CPU0:CRS# edit route-policy RP1 ?
  emacs  to use Micro Emacs editor
  inline to use command line
  nano   to use nano editor
  vim    to use Vim editor
  <cr>
```

© 2014 Cisco Systems, Inc. All rights reserved.

25

Use the **edit** command in EXEC mode to start editing a configuration object. Select the preferred editor.

The built-in editors are available for route policies and other objects, such as various sets that are used in combination with route policies.

The following list contains some of the most commonly used keystroke combinations within the Emacs editor:

- **Ctrl-F**: Move cursor forward (right).
- **Ctrl-B**: Move cursor backward (left).
- **Ctrl-N**: Move cursor to next line (down).
- **Ctrl-P**: Move cursor to previous line (up).
- **Ctrl-E**: Move to the end of the line.
- **Ctrl-A**: Move to the start of the line.
- **Backspace**: Delete character to the left of the cursor.
- **Ctrl-D**: Delete character to the right.
- **Ctrl-X** followed by **Ctrl-S**: Save changes.
- **Ctrl-X** followed by **Ctrl-C**: Exit and commit saved changes.

The following list contains some of the most commonly used commands within the VIM editor:

- **Left, down, up, and right** arrows: Move cursor left, down, up, right.
- **h, j, k, l**: Move cursor left, down, up, right.
- **i**: Start editing at the cursor position.
- **a**: Start editing after the cursor position.

- **ESC** : Stop editing (return to command mode).
- **x** : Delete the character at cursor position.
- **dd** : Delete a line.
- **u** : Undo a single action.
- **ESC** followed by **:w** : Save changes.
- **ESC** followed by **:q** : Exit and commit saved changes.

After exiting the editor, you will be asked to save and commit the changes.

Refer to the Cisco IOS XR Software command reference for a detailed list of all commands and options for all the available editors.

Value Sets

Value sets are objects that are used to modularize routing policies. Various types of sets exist for different types of parameters and attributes.

Value Sets

RPL can match attributes against a set of multiple values:

- Inline sets using parentheses for one-time use
- Named value sets for reusability

Value sets:

- AS path in AS path set
- Standard community in community set
- Extended community in extcommunity set
- Prefix in prefix set
- Route distinguisher in route distinguisher set

© 2014 Cisco Systems, Inc. All rights reserved.

76

Value Sets (Cont)

Inline value set

```
route-policy RP
  if attribute in (value, value2, ...)
  then
    set local-preference 200
  endif
end-policy
```

as-path-set
community-set
extcommunity-set
prefix-set
rd-set

Named value set

```
xy-set set-name
  value,
  value2
end-set

route-policy RP
  if attr in set-name then
    set local-preference 200
  endif
end-policy
```

© 2014 Cisco Systems, Inc. All rights reserved.

77

Each set can contain multiple values. The **in** operator can be used for the existence of a value in the set.

The example on the left illustrates a generic condition where an inline value set is used. The example on the right references a preconfigured value set.

AS Path Sets

An AS path set can contain one or more regular expressions. A condition can be used to check for an AS path attribute against the set of regular expressions.

AS Path Sets

AS path set characteristics:

- Define an AS path set using the **as-path-set** command.
- Use one or more comma-separated **ios-regex** commands to define the regular expressions that define set membership.
- Use the **in** operator in the routing policy to test for membership of the AS path in the AS path set.

Match prefixes originating in defined autonomous systems.

```
as-path-set PreferredOriginators
ios-regex '_10$',
ios-regex '_20$',
ios-regex '_30$',
ios-regex '_40$'
end-set
```

Use an AS path set in a policy to match prefixes based on AS path attribute.

```
route-policy RP
if as-path in PreferredOriginators then
set local-preference 200
endif
end-policy
```

© 2014 Cisco and/or its affiliates. All rights reserved. 75

The sample configuration uses a policy to set the local preference to 200 for all preferred originating autonomous systems that are listed in the AS path set.

AS Path Sets (Cont.)

Predefined Matching Criteria	Description
is-local	Matches any prefix with an empty AS path attribute (equals regular expression '^\$')
neighbor-is path	Matches based on first ASN in the AS path attribute (equals regular expression '^path_')
originates-from path	Matches based on last ASN in the AS path attribute (equals regular expression '_path\$')
passes-through ASN	Matches based on ASN anywhere in the AS path (equals regular expression '_path_')
length len	Matches AS paths based on number of ASNs in the path
unique-length len	Matches AS paths based on number of unique ASNs in the path

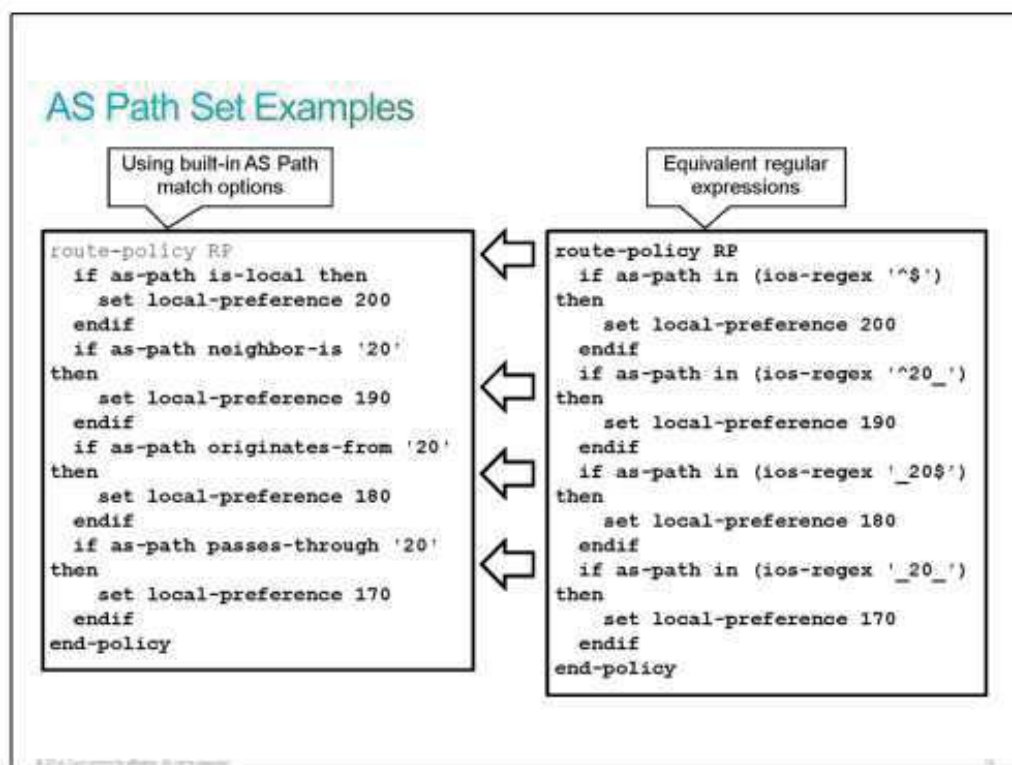
© 2014 Cisco and/or its affiliates. All rights reserved.

71

Instead of using regular expressions, some of the more common AS path checks can be performed using built-in conditions:

- **is-local** : Identifies whether a prefix is local to the AS; it performs the same function as a regular expression checking for an empty AS path attribute ("^\$").
- **neighbor-is path** : Identifies whether a prefix was received from a neighboring AS; equals regular expression 'path_'
- **originates-from path** : Identifies whether a prefix was originated by a specified AS; equals regular expression '_path\$'
- **passes-through ASN** : Identifies whether a prefix passed through the specified AS; equals regular expression '_ASN_'
- **length len** : Matches AS paths based on number of ASNs in the path
- **unique-length len** : Matches AS paths based on number of unique ASNs in the path

AS Path Set Examples



The two samples show configurations equivalent in result but using different approaches:

- A. The left-side sample uses built-in conditions.
- B. The right-side example uses regular expressions.

Notice the following in the example:

- The regular expression '^\$' can be replaced by the built-in operator **is-local**.
- The regular expression '^20_' can be replaced by the built-in operator **neighbor-is**.
- The regular expression '_20\$' can be replaced by the built-in operator **originates-from**.
- The regular expression '_20_' can be replaced by the built-in operator **passes-through**.

Standard Community Sets

Multiple BGP communities can also be grouped into a community set.

Standard Community Sets

Standard community sets characteristics:

- Define a standard community set using the **community-set** command.
- Use one or more comma-separated match options:
 - **ios-regex** commands to define regular expressions that define set membership
 - numbered membership matching
 - membership matching using well-known standard communities
- Use the **matches-any** operator to match routes that have at least one community in the community set.
- Use the **matches-every** operator in routing policy to match routes that have all communities in the community set.

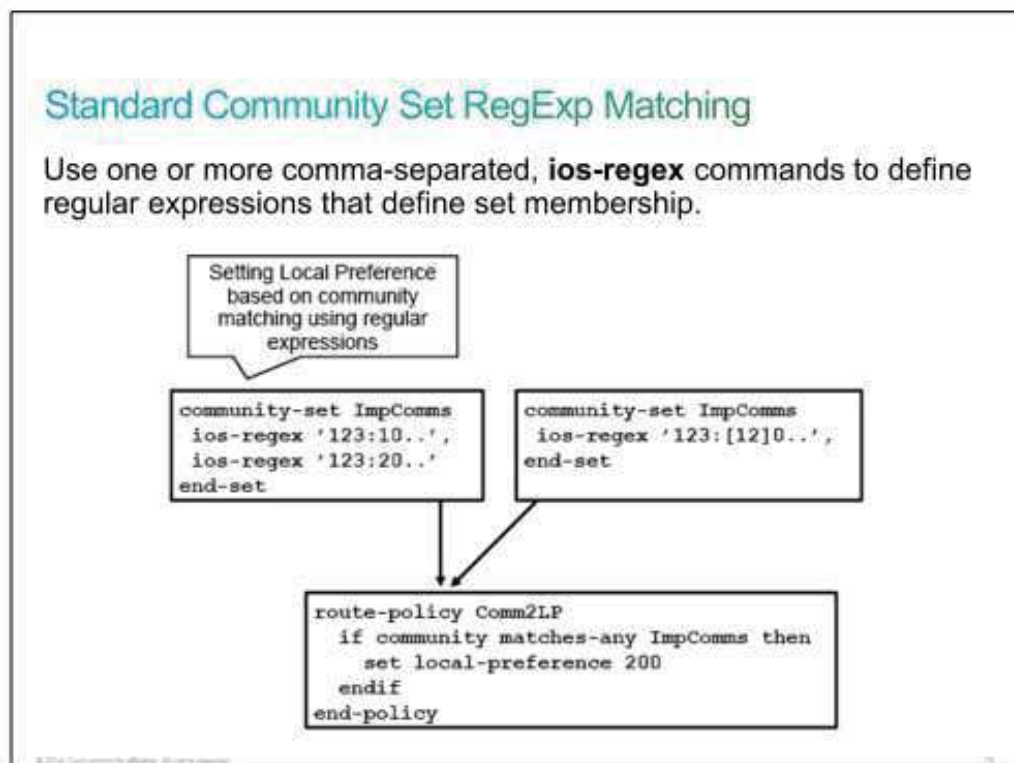
Communities can be matched in the following ways:

- Regular expression matching, where a regular expression is used against an ordered list of communities and converted into a string of characters.
- Numbered matching, where community attributes are matched against a list of values in a community set.
- Named matching, where community attributes are matched against a list of communities, including named well-known communities.

Community matching can use modifiers that define how the matching is performed:

- The **matches-any** operator should be used to match routes that have at least one community in the community set.
- The **matches-every** operator should be used to match routes that have all communities listed in the community set.

Standard Community Set RegExp Matching



The sample configuration illustrates two community sets, based on regular expressions.

The left-side ImpComms community-set uses two regular expressions; the right-side ImpComms community-set uses a single regular expression. Either approach can be used in the Comm2LP route-policy so that a route will be assigned local preference 200 if it contains BGP community 123:10xx or 123:20xx.

Standard Community Set Numbered Matching

Standard Community Set Numbered Matching

Use numbered matching:

- *AS:num*
- *AS:[range]*
- *AS:**

Setting Local Preference
based on numbered
community matching

```
community-set ImpComms
123:1010
123:[2000..2099]
999:*
end-set
!
route-policy Com2LP
  if community matches-any ImpComms then
    set local-preference 200
  endif
end-policy
```

The sample configuration illustrates numbered matching where explicit BGP communities are matched. Additionally, ranges and wildcards can be used in sets:

- “*AS:num*” is used to match a specific community.
- “*AS:[range]*” is used to match a range of values.
- “*AS:**” is used to match all values for a given *AS*.

The sample community set tries to match at least one community from a range that is defined using different options.

Standard Community Set Named Matching

Standard Community Set Named Matching

Use identifiers for well-known communities:

- **internet**: Match all communities.
- **local-as**: Keep tagged prefixes in the local AS.
- **no-advertise**: Prevent tagged prefixes from being advertised to any peer.
- **no-export**: Prevent tagged prefixes from being announced to EBGPeers.

Prevent sending of core subnets to external peers.

```
router bgp 1
 address-family ipv4 unicast
  redistribute connected route-policy
  NoExport
!
route-policy NoExport
 set community no-export
!
```

Delete all communities on incoming updates.

```
route-policy DeleteAllComms
 delete community in (internet)
end-policy
!
router bgp 1
 neighbor 1.2.3.4
  address-family ipv4 unicast
  route-policy DeleteAllComms in
!
```

The figure illustrates the third matching option for BGP communities; it is based on the names of well-known communities.

The left-side example assigns the **no-export** community to all redistributed routes. The right-side example matches all communities by using the **internet** keyword, and deletes them.

Prefix Sets

A prefix set is used to match routes based on prefix-list-like criteria in a prefix set.

Prefix Sets

Used to match prefixes in routing protocol updates: **prefix[/length
[{le | ge | eq} mask-len]]**

Various prefix sets

```
prefix-set PrivatePrefixes
 10.0.0.0/8 le 32,
 172.16.0.0/12 le 32,
 192.168.0.0/16 le 32
end-set
!
prefix-set CoreLoopbacks
 172.16.1.0/24 eq 32
end-set
!
prefix-set HostRoutes
 0.0.0.0/0 eq 32
end-set
```

Various prefix sets

```
prefix-set DefaultRoute
 0.0.0.0/0
end-set
!
prefix-set AllPrefixes
 0.0.0.0/0 le 32
end-set
!
prefix-set SmallPrefixes
 0.0.0.0/0 ge 24
end-set
!
prefix-set
SmallPrefixesExceptHostRoutes
 0.0.0.0/0 ge 24 le 31
end-set
```

The same syntax is used as with prefix lists.

Monitoring Routing Policies

The **show rpl** command with the **detail** keyword can be used to display the policy configuration, including all the dependencies.

Monitoring Routing Policies

Monitoring routing policies:

- Use the **show rpl route-policy [policy-name] [detail]** commands to display the policies.
- The detailed output also displays all referenced objects (e.g. sets and nested route policies).

RP/0/0/F00/CPU0:CRS# show rpl route-policy MgmtRTExport detail

```
extcommunity-set rt MgmtRT
  23456:100,
  12345:200
end-set
!
prefix-set MgmtLoopbacks
  10.1.1.0/24 le 32
end-set
!
route-policy MgmtRTExport
  if destination in MgmtLoopbacks then
    set extcommunity rt MgmtRT
  endif
end-policy
!
```

Display a policy and all other associated objects.

In the figure, the output shows the configurations of the MgmtRTExport route-policy as well as the configurations of the prefix set and the extended community-set referenced within the route policy.

Determining Attach Points

Determining Attach Points

Determining attach points:

- Use the **show rpl route-policy policy-name attachpoints** commands to list the attach points of the policy.
- The detailed output also displays all referenced objects (e.g. sets and nested route policies).

Display attach points
for a routing policy.

```
RP/0/RSP0/CPU0:PE1# show rpl route-policy pass attachpoints
Thu Nov 17 19:50:52.025 UTC
```

BGP Attachpoint: Neighbor

Neighbor/Group	type	afi/safi	in/out	vrf name
192.168.101.11	--	IPv4/unl	in	default
192.168.101.11	--	IPv4/unl	out	default

The **attachpoints** option can be used to display all references to the specified policy.

In the example, the **show** command shows that the specified route policy **pass** is attached to the **BGP** neighbor in the "in" and "out" direction.

Testing Routing Policies

Policies can be combined with the **show bgp** command to display only those **BGP** entries that are permitted by the policy.

Testing Routing Policies

Some policies can be tested (e.g. outbound BGP filter):

- Use the **show bgp route-policy policy-name** command to list BGP entries permitted by the policy.
- Note: Attributes modified by the policy are not displayed.

Test a new policy to filter outgoing updates.

```
RP/0/RP1/CFG0:CRS# show bgp route-policy FilterOut
BGP router identifier 0.0.0.0, local AS number 1
BGP generic scan interval 60 secs
BGP table state: Active
BGP main routing table version 30
BGP scan interval 60 secs
Status codes: s suppressed, d damped, h history, * valid, > best
              i - internal, S stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop           Metric LocPrf Weight Path
*> 10.4.100.0/30   0.0.0.0                 0      200   32768 ?

Processed 1 prefixes, 1 paths
RP/0/RP1/CFG0:CRS#
```

This approach can be used to test the performance of a newly configured policy or to limit the display of a large BGP table for troubleshooting purposes.

In the example, the policy FilterOut only displays one entry (10.4.100.0/30) in the BGP table.

Translating Route Maps to Routing Policies

When you translate route maps to routing policies, understanding the relationship between multiple conditions in a single route map statement is important.

Translating Route Maps to Routing Policies

When you migrate from Cisco IOS or IOS XE Software to Cisco IOS XR Software, use the following guidelines to translate route maps to policies:

- Each numbered entry is one **if** statement.
- Each **match** option is one condition:
 - Match conditions of the same type should be joined using the **OR** logical operator.
 - Match conditions of different types should be joined using the **AND** logical operator.
 - Use parentheses to maintain proper precedence.

Recall that multiple conditions of the same type are combined by using a logical **OR**. Therefore, you should use the **OR** operator in the **if** statement of the routing policy. Multiple conditions of different types are combined using a logical **AND**. Therefore, you should use the **AND** operator in the **if** statement of the routing policy. Make sure that you use parentheses for proper operator precedence.

Translating Route Maps to Routing Policies (Cont.)

```
route-map RM permit 10
  match ip address prefix-list PL1
  set local-preference 200
!
route-map RM permit 20
  match ip address prefix-list PL2
  set local-preference 150
!
```

Sample route map

Translated routing
policy

```
route-policy RP
  if destination in PL-Set1 then
    set local-preference 200
  elseif destination in PL-Set2 then
    set local-preference 150
  endif
end-policy
```

The two sample configurations show how two route map statements can be translated into a routing policy.

Summary

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

Summary

- Route maps are a simple language to support complex routing policies.
- A route map is processed in a top-down fashion. There is implicit deny at the end of a route map.
- The RPL is a newer mechanism that was introduced into Cisco IOS XR Software as a replacement and improvement upon the route maps.
- In Cisco IOS XR Software, BGP updates are not forwarded to an external neighbor unless an outbound policy is attached to the neighbor.
- You can apply routing policies to many attach points, such as redistribution between any pair of routing protocols.
- Value sets are objects that are used to modularize routing policies. Various types of sets exist for different types of parameters and attributes.
- You can use a prefix set to match routes based on prefix-list-like criteria.

© 2014 Cisco and/or its affiliates. All rights reserved.

93

Lesson 3: Implementing Route Redistribution

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. Understanding how to exchange routing information between these routing protocols, and how Cisco routers operate in an environment with multiple routing protocols is important.

Configuring route redistribution can be simple or complex, depending on the mix of routing protocols that you want to redistribute. The commands that are used to enable redistribution and assign metrics vary slightly, depending on the routing protocols that are 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 examines several reasons for using more than one routing protocol, describes redistribution from one routing protocol to another, and explains how Cisco routers make route selections when multiple protocols are active in the network. This lesson also describes how to configure route redistribution between various routing protocols. The commands for each protocol are covered. In addition, the impact of route redistribution is analyzed.

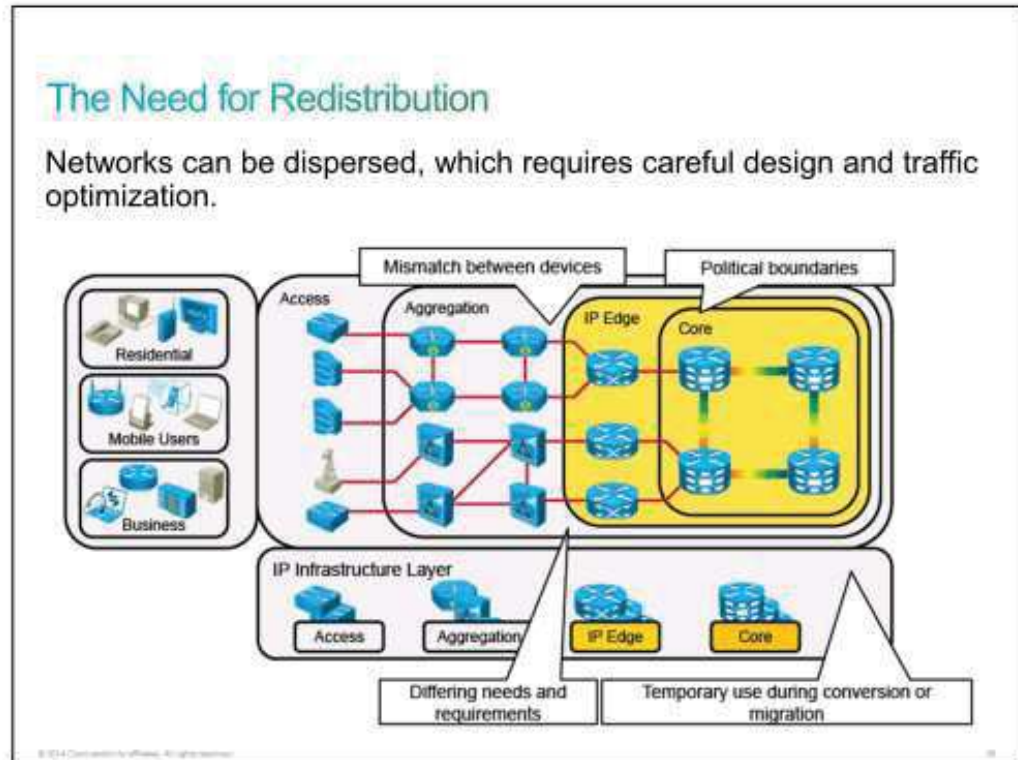
Upon completing this lesson, you will be able to:

- Describe the need to use redistribution in the multiple IP routing protocol environment
- Describe route redistribution
- Describe using seed metrics in route redistribution
- List the default seed metrics
- Describe redistribution on a single router
- Describe redistribution on multiple devices that are running both routing protocols
- Describe techniques for performing route redistribution

- Describe potential problems with redistributing routes and identify potential solutions
- Describe the procedures that are necessary to configure route redistribution
- Describe how to redistribute routes into OSPF
- Describe how to redistribute routes into IS-IS
- Describe how to redistribute routes into BGP
- Describe administrative distance
- Describe how to modify the administrative distance for OSPF routes
- Describe how to modify the administrative distance for IS-IS and BGP routes
- Show an example of using the administrative distance to modify route redistribution
- Describe how to prevent routing loops with multipoint two-way redistribution

The Need for Redistribution

There are many reasons for complex routing scenarios. When a network grows, it typically becomes more complex as network administrators adjust it based on factors such as political borders, geographical borders, and mergers with other companies.



In all such situations, network administrators face complex routing scenarios, in which the use of multiple IP routing protocols is not unusual. With large numbers of routers and routing protocols running, the number of routing updates increases. As the network grows larger, traffic from those updates can slow down the network, indicating that a change is required.

There are many possible solutions when the design of a network must be changed or optimized. A scalable routing protocol may be necessary, requiring careful design and traffic optimizations, such as the following:

- Redistribution
- Routing traffic filtering
- Summarization

Multiple routing protocols may be necessary in situations such as the following:

- When you are migrating from an older IGP to a new IGP, multiple redistribution boundaries may exist until the new protocol has completely displaced the old protocol. The same applies to company mergers between companies that are each using a different routing protocol.
- Although you want or need a new protocol, the old routing protocol may still be needed for host systems, such as UNIX host-based routers that are running RIP.
- Some departments do not want to upgrade their routers to support a new routing protocol.

- You work with mixed-router vendor environments. In these environments, you can use a routing protocol that is specific to Cisco, such as EIGRP, in the Cisco portion of the network; you can use 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 this is not always the best policy. The network design may not allow default routes.

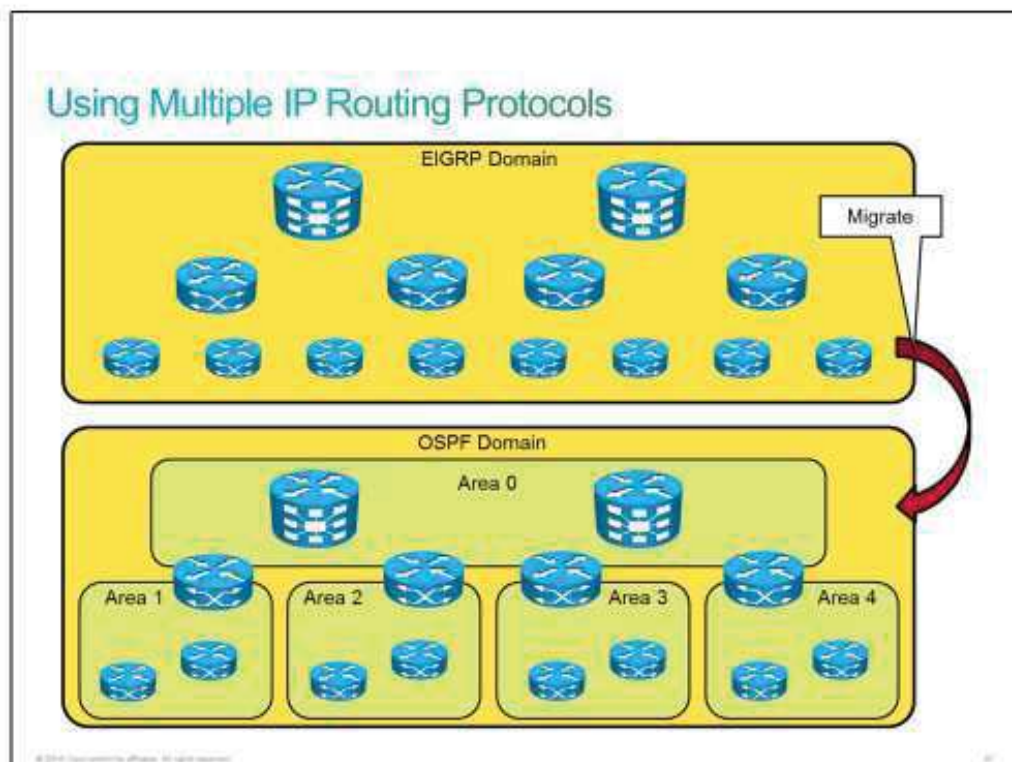
If there is more than one way to get to a destination network, routers may need information about routes in other parts of the network to determine the best path to that destination. Additionally, if there are multiple paths, a router must have sufficient information to determine a loop-free path to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

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

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

Using Multiple IP Routing Protocols

Whatever the reason for running multiple protocols, network administrators must conduct the migration from one routing protocol to another, or the redistribution of routing information between them, carefully and thoughtfully.



Network administrators must understand what must be changed, and 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.

Network administrators must keep in mind the requirements and capabilities of differing routing protocols when they are planning redistribution. Link-state routing protocols, such as OSPF, require a hierarchical network structure. Decisions must be made about 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. Network administrators must carefully plan the redistribution strategy to avoid disrupting network traffic or causing outages.

Route Redistribution

Redistribution is the process of using a routing protocol to advertise routes that are learned by the usual means of learning routes, such as by another routing protocol, static routes, or directly connected routes.

Route Redistribution

Redistribution characteristics:

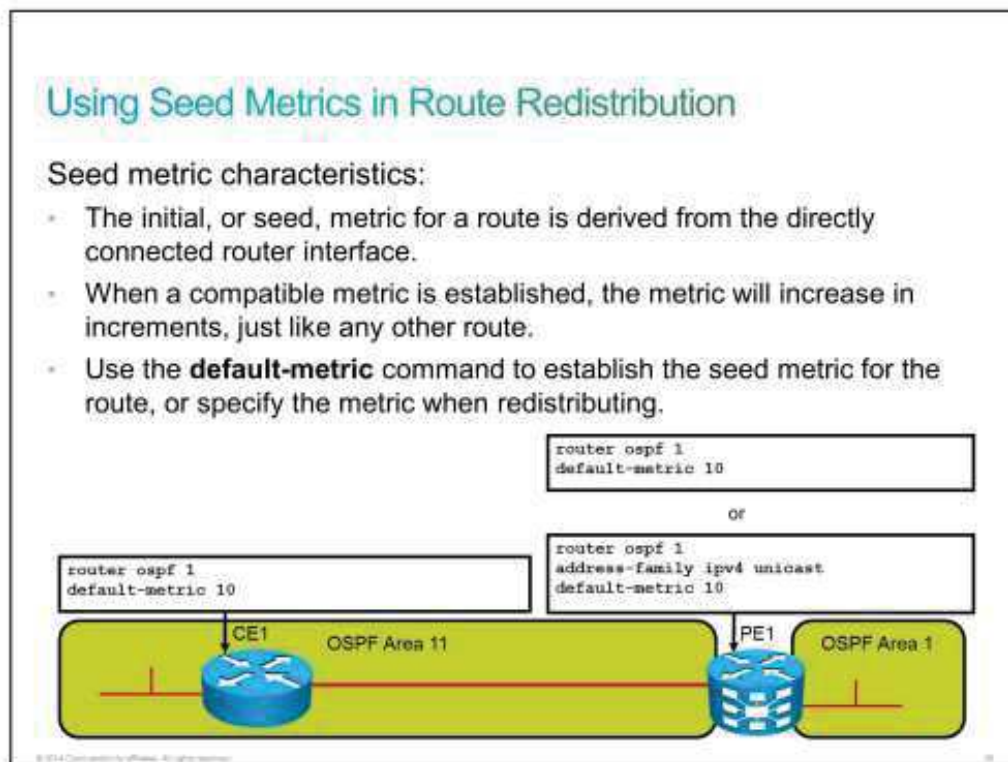
- Routes learned by some other means are selectively redistributed into a routing protocol from one of three sources:
 - Another routing protocol
 - Static routes
 - Directly connected routes
- Routing loop prevention:
 - Only routes used by the router itself are redistributed.

While it is desirable that you run a single routing protocol throughout your entire IP internetwork, multiprotocol routing is common for many reasons. These reasons include company mergers, multiple departments that are managed by multiple network administrators, and multivendor environments. Running different routing protocols is often part of a network design. Whatever the reason, if you have a multiprotocol environment, redistribution is a necessity.

To have a scalable solution and limit the amount of routing update traffic, the redistribution process must selectively insert the routes that are learned. Redistribution can lead to routing loops, which must be avoided. Only routes that are used by the router itself should be redistributed.

Using Seed Metrics in Route Redistribution

When a router advertises a link that is directly connected to one of its interfaces, the initial metric (also called the seed metric or default metric) is derived from the characteristics of that interface; this metric increases in increments as the routing information is passed to other routers.



For OSPF, the seed metric is based on the bandwidth of the interface. For EGRP, the seed metric is based on the interface bandwidth and delay. For RIP, the seed metric starts with a hop count of 0, and increases in increments from router to router.

Redistributed routes are not physically connected to a router; they are learned from other routing protocols. It is difficult to translate from one metric to another, that is, from hops to bandwidth. If a boundary router is to redistribute information between routing protocols, it must be able to translate the metric of one routing protocol into the metric of the other routing protocol.

For example, if a boundary router receives an RIP route, the route will have a hop count as its metric. To redistribute the route into OSPF, the router must translate the hop count into a cost metric that the OSPF routers will understand. This seed metric is defined during redistribution configuration. Once the seed metric for a redistributed route is established, the metric will increase in increments, normally within the AS.

There is an exception to the rule that metrics increase in normal increments within the AS. OSPF E2 routes maintain their initial metric, regardless of how far they are propagated across an AS.

The Cisco IOS or IOS XE or IOS XR **default-metric** command, which is used in 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, by using a route map, or RPL.

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

Default Seed Metrics in Route Redistribution

The figure below shows the default seed metric value for redistributed routes for each IP routing protocol.

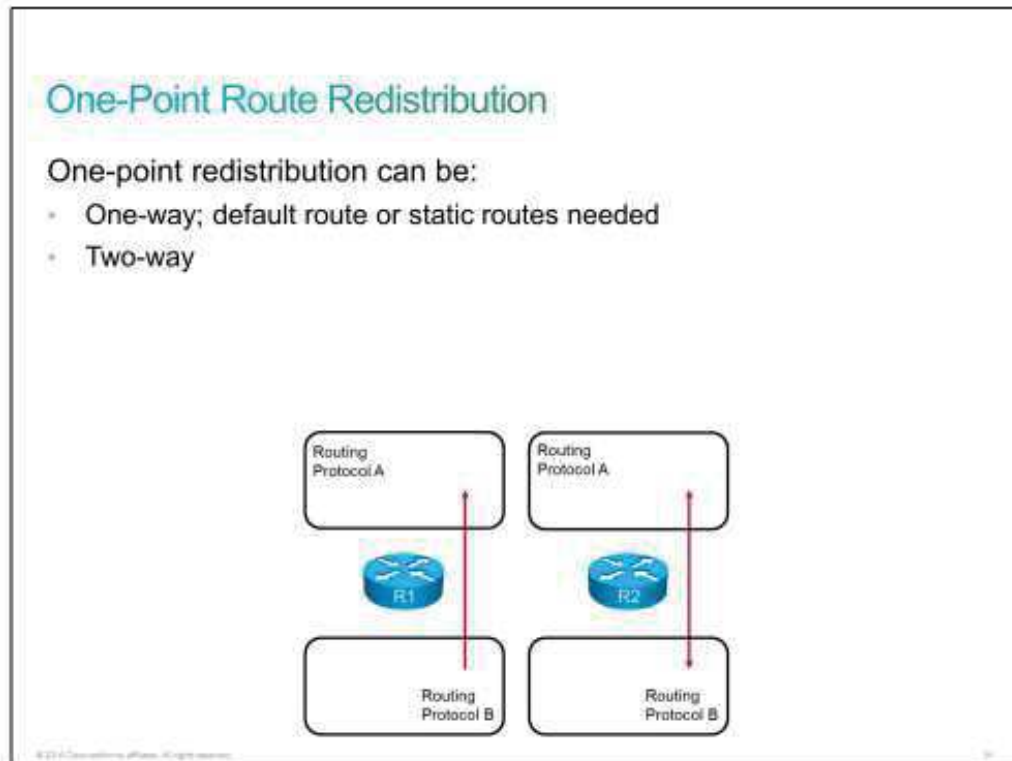
Default Seed Metrics in Route Redistribution					
Source	Into RIP	Into EIGRP	Into OSPF	Into IS-IS	Into BGP (MED)
Connected	1	Interface metric	20 (E2)	0	0
Static	1	Interface metric	20 (E2)	0	0
RIP		Infinite	20 (E2)	0	IGP metric
EIGRP	Infinite		20 (E2)	0	IGP metric
OSPF	Infinite	Infinite		0	IGP metric
IS-IS	Infinite	Infinite	20 (E2)		IGP metric
BGP	Infinite	Infinite	1 (E2)	0	

RIP and EIGRP do not advertise a redistributed route unless a seed metric is configured.

These protocols interpret the seed metric of 0 as infinity by default. A metric of infinity tells the router that the route is unreachable; therefore, it should not be advertised.

One-Point Route Redistribution

One-point redistribution defines only one redistribution point between two routing protocols. Routes are redistributed on one router only.



The one-point redistribution can be one of two types:

- One-way
- Two-way

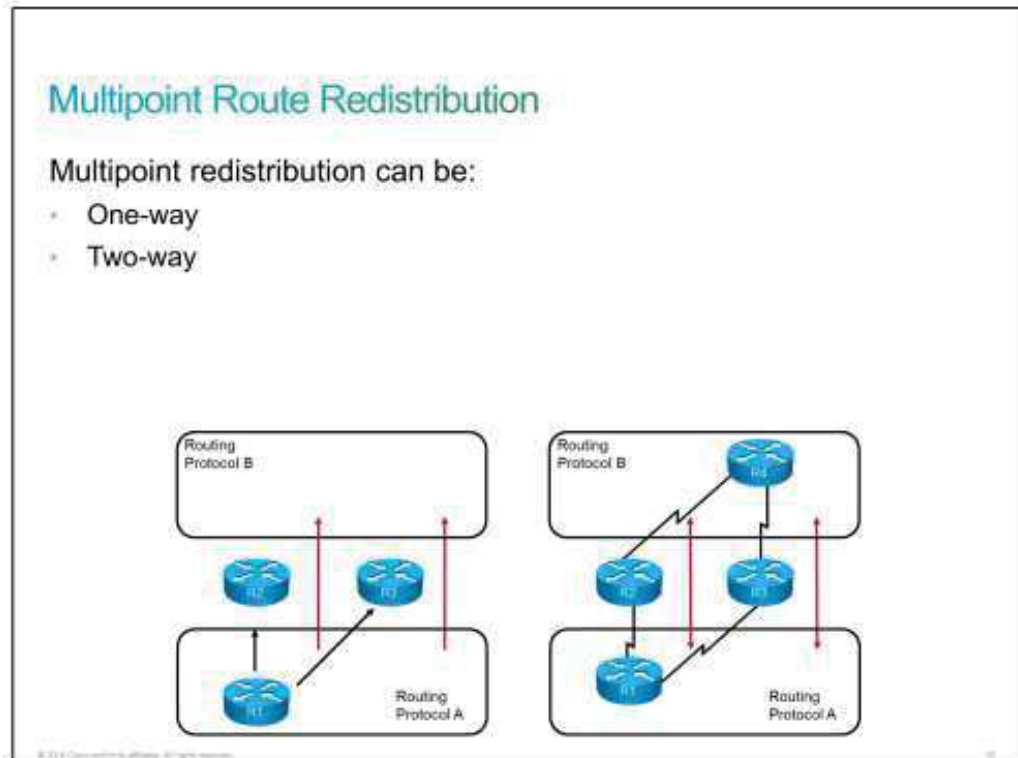
Redistribution is one-way if routes from routing protocol A are redistributed into routing protocol B, but not vice versa. Redistribution is two-way if routes from routing protocol A are redistributed into routing protocol B, and routes from routing protocol B are also redistributed into routing protocol A.

One-way redistribution requires the use of a default route or static routes. If routes are redistributed from routing protocol A into routing protocol B, routing protocol B devices are aware of all the routing information. At the same time, devices in the routing protocol A autonomous system are aware of routing information for their AS only, and reachability for destinations that are outside the routing protocol A autonomous system requires the use of a default route or one or more static routes.

One-way or two-way redistribution at one point is always safe, because one-point redistribution represents the only exit and entrance from one routing protocol to another. Routing loops cannot be inadvertently created.

Multipoint Route Redistribution

Multipoint redistribution is redistribution between two routing protocols that takes place on two or more separate devices that are running both routing protocols.



Two possibilities of multipoint redistribution exist:

- Multipoint one-way redistribution
- Multipoint two-way redistribution

Multipoint redistribution is likely to introduce routing loops. Even one-way multipoint redistribution is problematic, and generic multipoint two-way redistribution is highly problematic. Problems often result from differences in administrative distance between the two protocols, and from incompatible metrics. Statically assigned metrics are used in redistribution points.

Multipoint one-way redistribution only works well under these circumstances:

- The receiving routing protocol supports different administrative distances for internal and external routes. Routing protocols that support different administrative distances include [EIGRP](#), [BGP](#), and recent maintenance releases of [OSPF](#).
- The external administrative distance of the receiving routing protocol is higher than the administrative distance of the sending routing protocol.

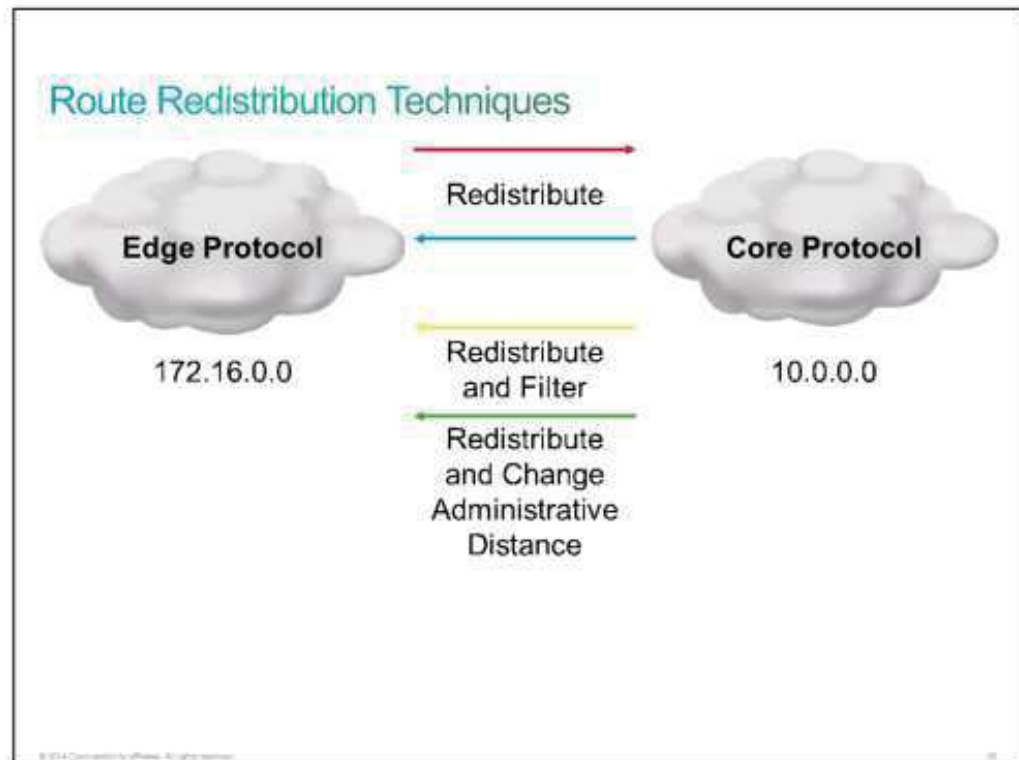
Multipoint two-way redistribution includes difficulties:

- Suboptimal routing (only part of the total cost is considered in routing decisions)
- Self-sustained routing loops on route loss

In multipoint redistribution scenarios, preventing routing loops is a main concern. The redistribution configuration should insert only internal routes from routing protocol A to B and vice versa. Routes at the redistribution points should be tagged and then filtered, based on the tags that are used when doing redistribution in the other direction. Propagation of the metric from A to B and vice versa is recommended, even though it is not sufficient to prevent loops. The easiest way to avoid loops when using two-way redistribution is to use a default route.

Route Redistribution Techniques

The safest way to perform redistribution is to redistribute routes in only one direction, on only one boundary router within the network. To do this, you must first determine which routing protocol is the core routing protocol, and which ones are edge routing protocols.



The core routing protocol is the main routing protocol that is running in the network. During a transition between routing protocols, the core is the new routing protocol and the edge is the old routing protocol. In networks that run multiple routing protocols all the time, the core is usually the more advanced routing protocol.

If redistribution must be done in both directions, or on multiple boundary routers, the redistribution should be tuned to avoid problems such as suboptimal routing and routing loops.

Depending on your network design, you may use any of the following redistribution techniques:

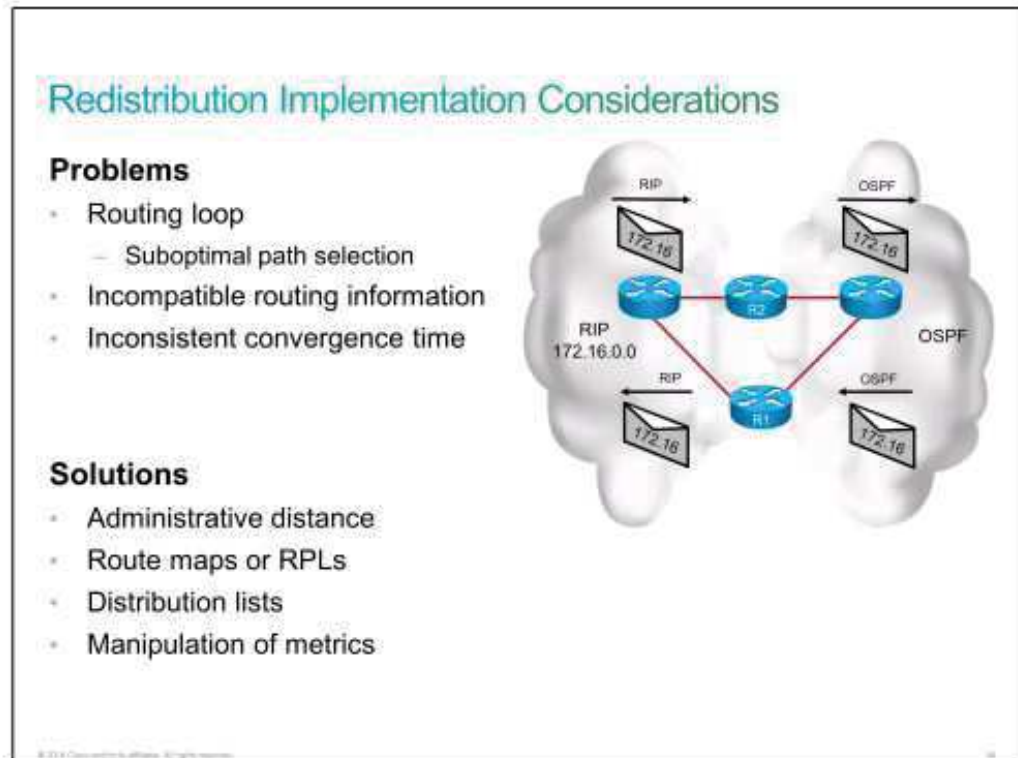
- **Redistribute a default route about the core AS into the edge AS:** In one-way redistribution, routes from the edge routing protocols are redistributed into the core routing protocol, and a default route is sent back to the edge routers. This technique helps prevent route feedback, suboptimal routing, and routing loops.
- **Redistribute multiple static routes about the core AS into the edge AS:** The edge routes are still redistributed into the core, but static routes for the core networks are redistributed into the edge protocol and sent to the edge routers. This method works if there is one redistribution point only, but it may cause route feedback if there are multiple points.
- **Redistribute routes from the core AS into the edge AS with filtering to block out inappropriate routes:** For example, routes from the edge should not be redistributed back into the edge routers from the core via another redistribution point (when there are multiple boundary routers).

- **Redistribute all routes from the core AS into the edge AS and from the edge AS into the core AS:** Then, modify the administrative distance that is associated with the external routes so that they are not the selected routes when multiple routes exist for the same destination. In some cases, the route that is learned by the native (local) routing protocol is better, but it may have a higher (less believable) administrative distance.

If two routing protocols advertise routes to the same destination, information from the routing protocol with the lowest administrative distance is placed in the routing table. By default, a route that is redistributed into a routing protocol inherits the default administrative distance of that routing protocol.

Redistribution Implementation Considerations

Redistribution of routing information adds to the complexity of a network and increases the potential for routing confusion, so it should be used only when necessary.

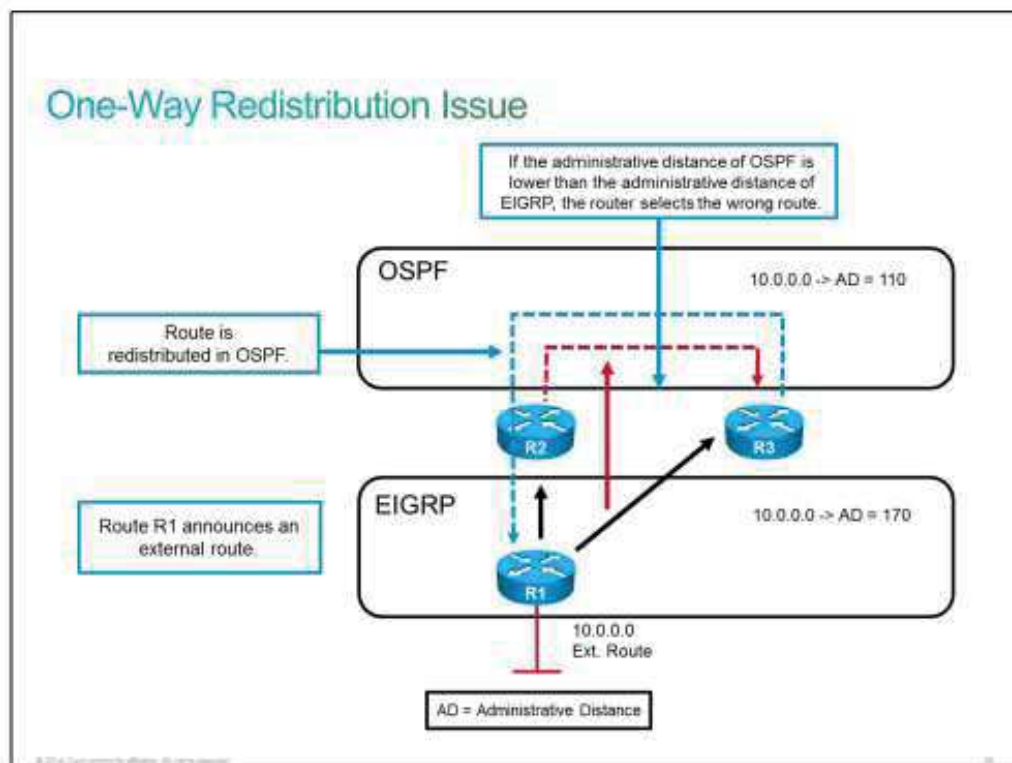


Several important issues arise when you are using redistribution:

- **Routing feedback (routing loops):** Depending on how you employ redistribution, routers may send routing information that is received from one AS back into that same AS. The feedback is similar to the routing loop problem that occurs in distance vector topologies.
- **Incompatible routing information:** Because routing protocols use different metrics to determine the best path, path selection using the redistributed route information may be suboptimal. The metric information about a route cannot be translated exactly into a different protocol, so the path that a router chooses may not be the best. Generally, to prevent suboptimal routing, you should assign to redistributed routes a seed metric that is higher than any routes that are native to the redistributing protocol. For instance, if RIP routes are being redistributed into OSPF and the highest OSPF metric is 50, the redistributed RIP routes should be assigned an OSPF metric that is higher than 50.
- **Inconsistent convergence time:** Different routing protocols converge at different rates. For example, RIP converges more slowly than EIGRP, so if a link goes down, the EIGRP network will learn about it before the RIP network does.

Good planning will ensure that these issues do not cause problems in your network. Good planning can eliminate the majority of issues, but additional configuration might be required. Some issues may be solved by changing the administrative distance, manipulating the metrics, and filtering using route maps, RPLs, and distribute lists.

One-Way Redistribution Issue



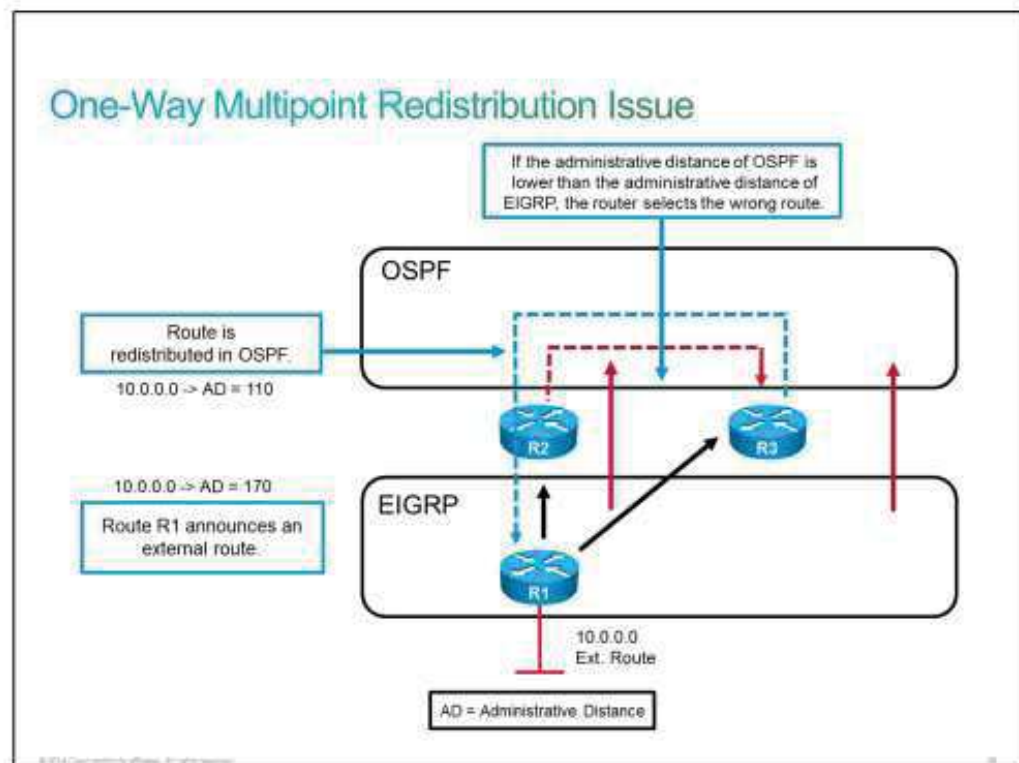
Multipoint one-way redistribution can cause routing confusion if routing protocols are using different administrative distances. It works properly only in these circumstances:

- If the receiving routing protocol supports different administrative distances for internal and external routes.
- If the external administrative distance of one routing protocol (OSPF in the figure) is higher than the administrative distance of the second routing protocol (EIGRP in the figure).

Routing protocols that support different administrative distances include EIGRP, BGP, and OSPF.

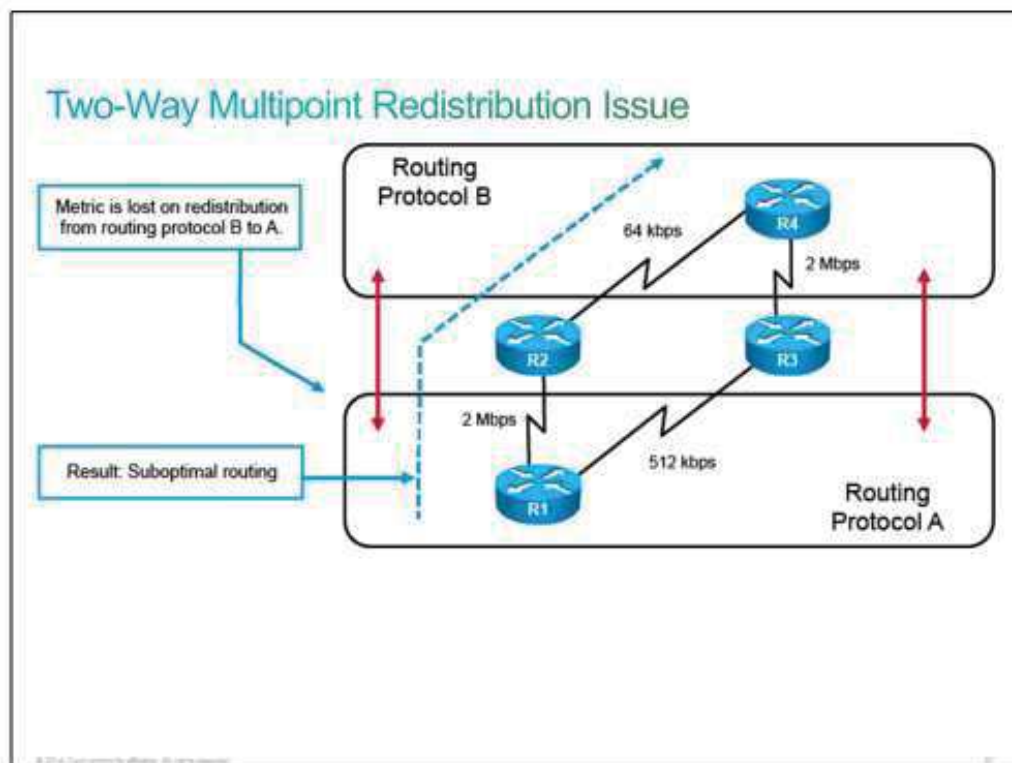
In the figure, R1 from EIGRP is announcing an external route by sending routing updates to R2 and R3. Both neighboring routers are running two routing protocols, and the redistribution between EIGRP and OSPF takes place on R2. R3 receives routing update information for the same route, directly from R1 and via R2, that is sending a redistributed route through OSPF updates. The administrative distance of OSPF (110) is lower than the administrative distance of external EIGRP routes (170), so R3 selects the wrong route. Instead of sending packets directly to R1, R3 prefers the path via R2, and the result is suboptimal routing.

One-Way Multipoint Redistribution Issue



In the figure, R1 from EIGRP is announcing routes by sending routing updates to R2 and R3. Both neighboring routers are running two routing protocols, and redistribution between EIGRP and OSPF takes place. Because multipoint redistribution is configured, R2 and R3 receive routing update information for the same route directly from R1 and via OSPF (R2 from R3 and R3 from R2). The routing update information that they receive via OSPF comes in the form of redistributed routes that they receive in OSPF updates. If the administrative distance of OSPF (110) is lower than the administrative distance of EIGRP (170), then R2 also selects the wrong route. Instead of sending packets directly to R1, R2 prefers the path via R3, and the result is suboptimal routing. It can also result in route flapping, because R2 is also redistributing the same EIGRP route in OSPF. R3 also prefers the OSPF path and sends packets via R2. The packet may bounce back and forth between R2 and R3 in the OSPF network until its TTL value reaches 0, before being discarded. Network 10.0.0.0 is then unreachable.

Two-Way Multipoint Redistribution Issue



Generic multipoint two-way redistribution is highly problematic and requires careful design and configuration.

Routing protocols have incompatible metrics, and during redistribution the metric information can be lost. To avoid problems that are associated with incompatible metrics, statically assigned metrics can be used in redistribution points.

A number of problems can occur during multipoint two-way redistribution:

- Suboptimal routing (only part of the total cost is considered in routing decisions)
- Self-sustained routing loops upon route loss

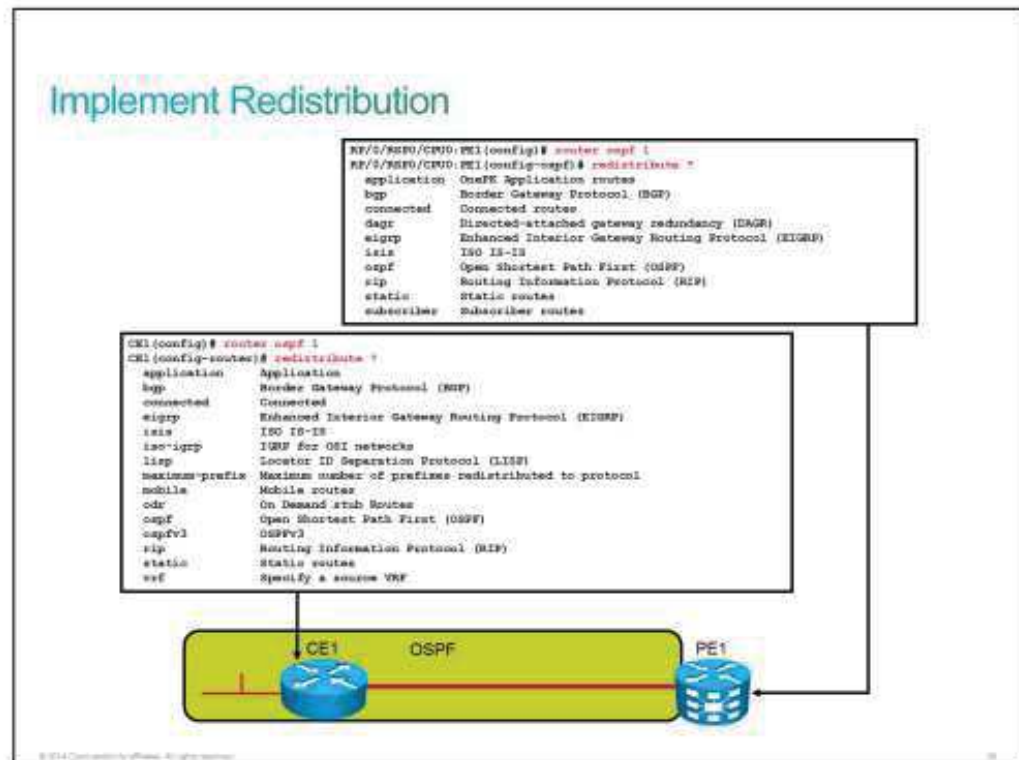
To prevent routing loops in multipoint redistribution scenarios, you should take into account the following recommendations during configuration:

- Insert only internal routes from routing protocol A to B and vice versa.
- Tag routes in redistribution points, and filter based on tags when doing redistribution.
- Propagate metrics from routing protocol A to routing protocol B properly (though this is not sufficient to prevent loops).
- Use default routes to avoid two-way redistribution.

The figure shows a two-way multipoint redistribution issue in which the cost of the internal links in routing protocol A is completely different from the cost of the links in routing protocol B. It is obvious that the best path between R1 and R4 is via R3, but during redistribution from routing protocol B to routing protocol A, the metric is lost and R1 sends the packets toward R4 via R2. The result is suboptimal routing.

Implement Redistribution

Redistribution supports all routing protocols, IPv4 and IPv6. Additionally, static and connected routes can be redistributed to allow the routing protocol to advertise the routes without using a network statement for them.

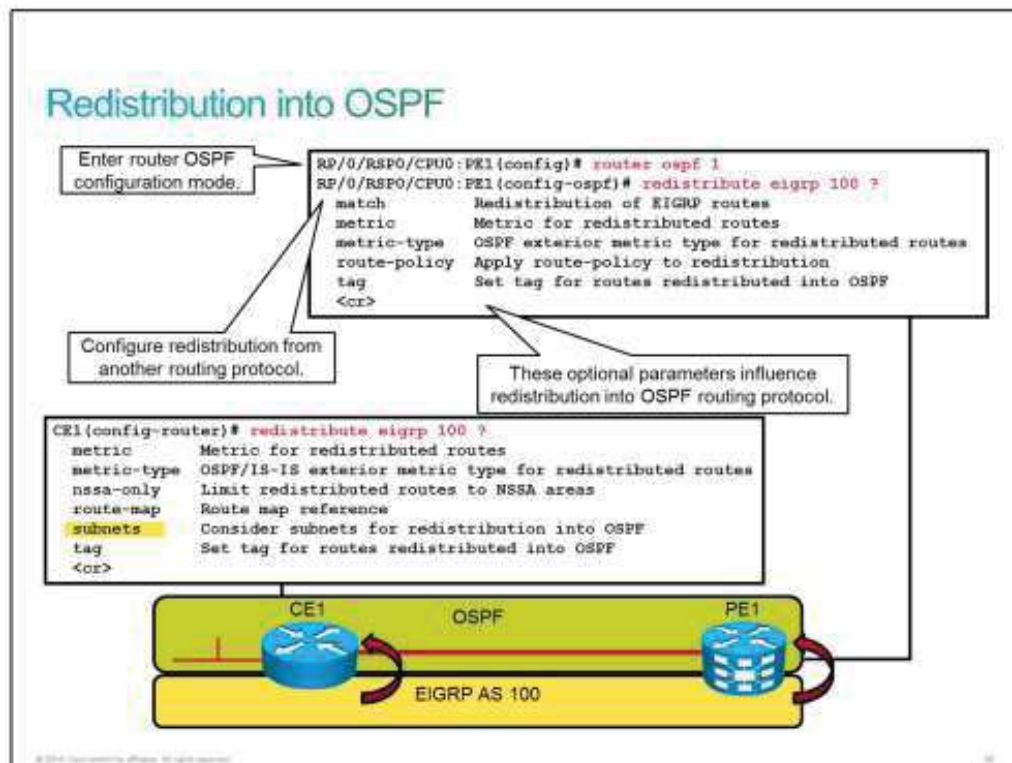


Routes are redistributed into a routing protocol, so the **redistribute** command is entered under the routing process that is to receive the routes. Before implementing redistribution, consider these points:

- Only protocols that support the same protocol stack are redistributed. For example, you can redistribute between IP RIP and OSPF, because they both support the TCP/IP stack or you can redistribute RIPng and OSPFv3 for IPv6 routes.
- The method that is used to configure redistribution varies slightly among different routing protocols and combinations of routing protocols. Some routing protocols require a metric to be configured during redistribution, but others do not.

Redistribution into OSPF

The figure shows how to configure for redistribution from EIGRP AS 100 into an OSPF routing process.



The first step is to enter the router OSPF configuration mode. The Cisco IOS or IOS XE or IOS XR **router ospf 1** command is used to access the OSPF routing process into which the routes need to be redistributed. In this case, it is OSPF process 1.

The next step is to use the Cisco IOS or IOS XE or IOS XR **redistribute** command to specify the routing protocol that is to be redistributed into OSPF. You can use optional keywords to change the way distribution is performed. For example, you might modify the default metric or route filtering using **route-policy** or **route-map**. The default metric is 20, and the default metric type is 2. In the Cisco IOS or IOS XE, subnets are not redistributed by default.

In OSPFv3, use the Cisco IOS or IOS XE or IOS XR **redistribute** command to redistribute from other routing protocols.

This example shows Cisco IOS XR Software output:

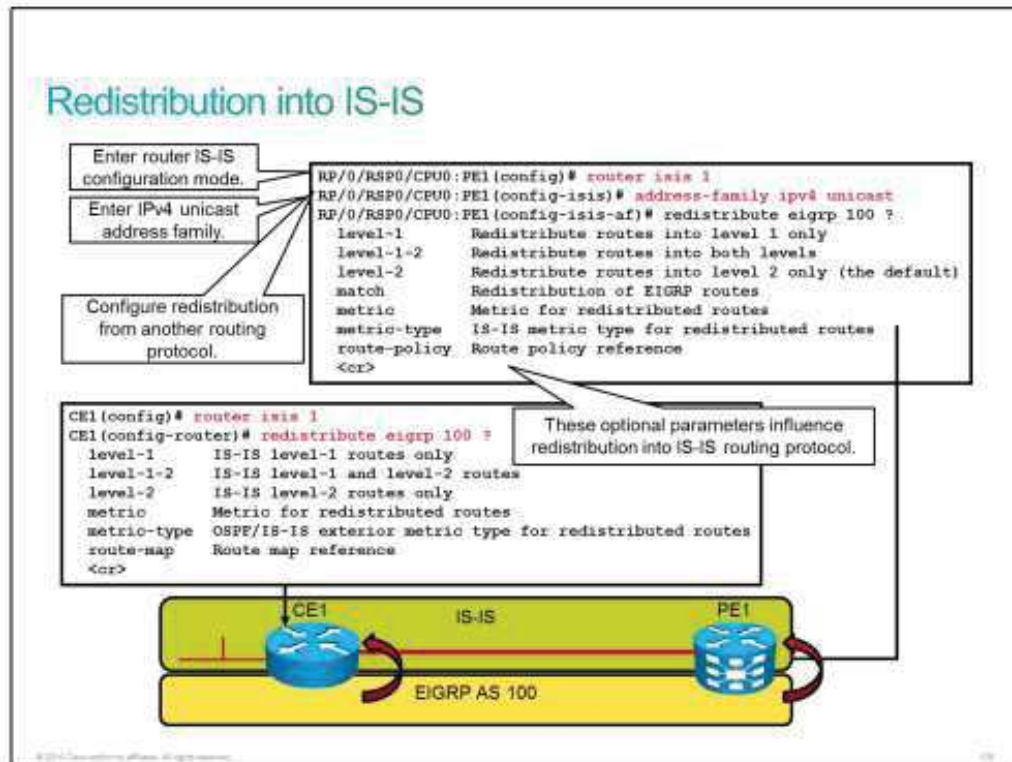
```
RP/0/RSP0/CPU0:PE1(config)# router ospfv3 1
RP/0/RSP0/CPU0:PE1(config-ospfv3)# redistribute ?
application  Application routes
bgp          BGP routes
connected    Connected routes
eigrp        Enhanced Interior Gateway Routing Protocol (EIGRP)
isis         IS-IS routes
ospfv3       OSPFv3 routes
static       Static routes
subscriber   Subscriber routes
```

This example shows Cisco IOS or IOS XE Software output:

```
CE1(config)# ipv6 router ospf 1
CE1(config-rtr)# redistribute ?
  application    Application Protocol
  bgp            Border Gateway Protocol (BGP)
  connected     Connected Routes
  eigrp         Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis          ISO IS-IS
  lisp          Locator ID Separation Protocol (LISP)
  nd            Neighbor Discovery
  nemo          Network Mobility (NEMO)
  ospf          Open Shortest Path First (OSPF)
  rip           IPv6 Routing Information Protocol (RIPv6)
  static        Static Routes
```

Redistribution into IS-IS

The figure shows how to configure for redistribution from EIGRP AS 100 into an IS-IS routing process.

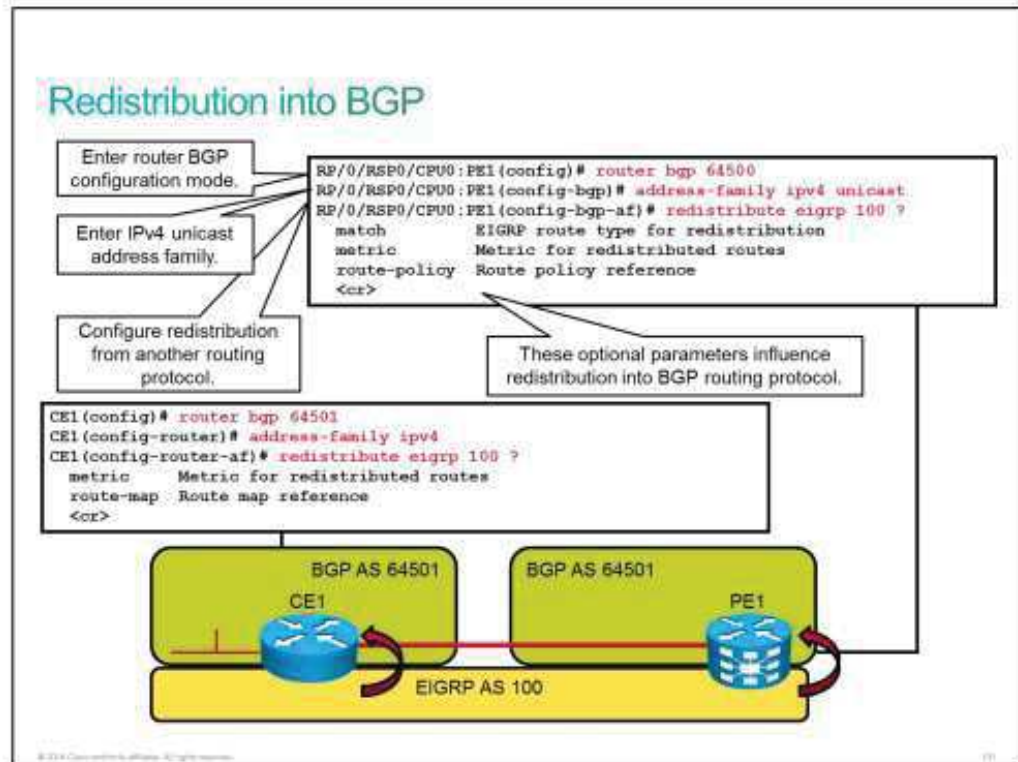


The first step is to enter the router IS-IS configuration mode. The Cisco IOS or IOS XE or IOS XR **router isis 1** command is used to access the IS-IS routing process into which the routes will be redistributed. On the Cisco IOS XR, use the **address-family ipv4 unicast** router IS-IS command to enter the address family for IPv4 unicast. On the Cisco IOS or IOS XE or IOS XR, use the **address-family ipv6 unicast** router IS-IS command to enter the address family for IPv6 unicast.

The next step is to use the Cisco IOS or IOS XE or IOS XR **redistribute** command to specify the routing protocol that is to be redistributed into IS-IS. You can use optional keywords to change the way distribution is performed. For example, you might modify the default metric or route filtering by using **route-policy** or **route-map**.

Redistribution into BGP

The figure shows how to configure for redistribution from EIGRP AS 100 into a BGP routing process.



The first step is to enter the router BGP configuration mode. The Cisco IOS or IOS XE or IOS XR **router bgp as_number** command is used to access the BGP routing process into which the routes will be redistributed. Use the Cisco IOS or IOS XE or IOS XR **address-family ipv4|ipv6 unicast** router BGP command to enter the address family for IPv4 or IPv6 unicast.

The next step is to use the Cisco IOS or IOS XE or IOS XR **redistribute** command to specify the routing protocol that is to be redistributed into BGP. You can use optional keywords to change the way distribution is performed. For example, you might modify the default metric or route filtering using **route-policy** (Cisco IOS XR) or **route-map** (Cisco IOS or IOS XE).

Administrative Distance

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

Administrative Distance	
• A lower administrative distance means a route is more trustworthy.	
Route source	Default AD
Connected interface	0
Static route	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
OSPF	110
IS-IS	115
RIPv1, RIPv2	120
External EIGRP	170
Internal BGP	200
Unknown	255

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 is a way of ranking the trustworthiness of a routing protocol. Cisco has assigned a default administrative distance value to each routing protocol that is supported on its routers.

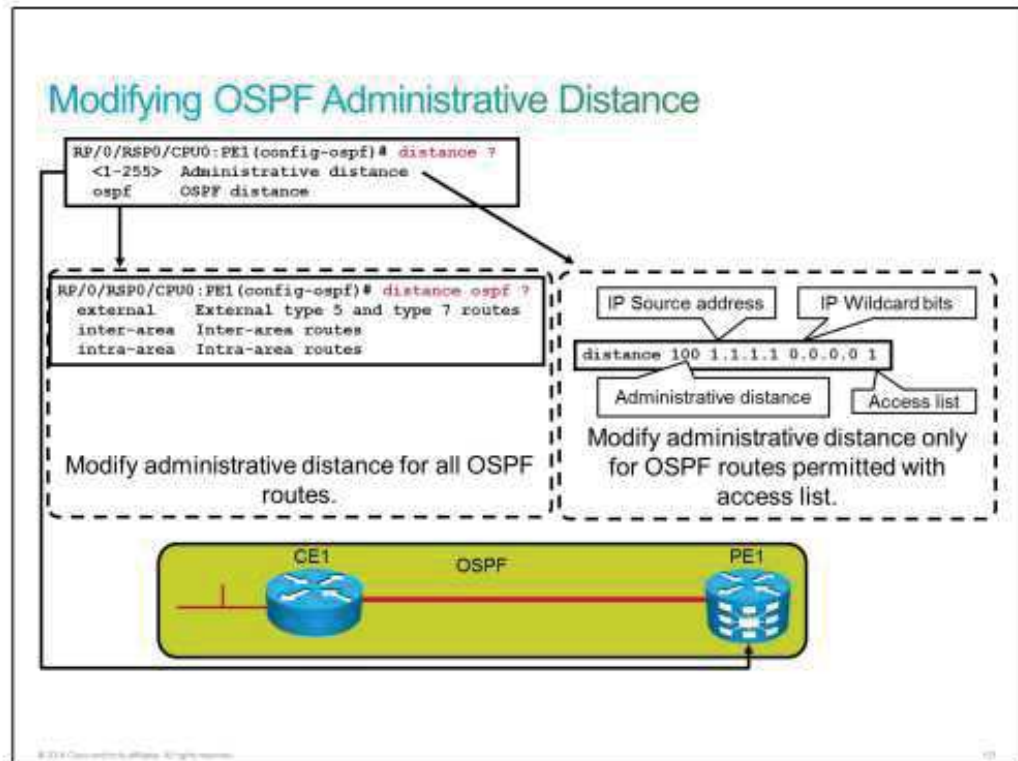
Each routing protocol is prioritized from most believable to least believable. Some examples of prioritization are as follows:

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

The table in the figure lists the default administrative distances of the protocols that are 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.

Modifying OSPF Administrative Distance

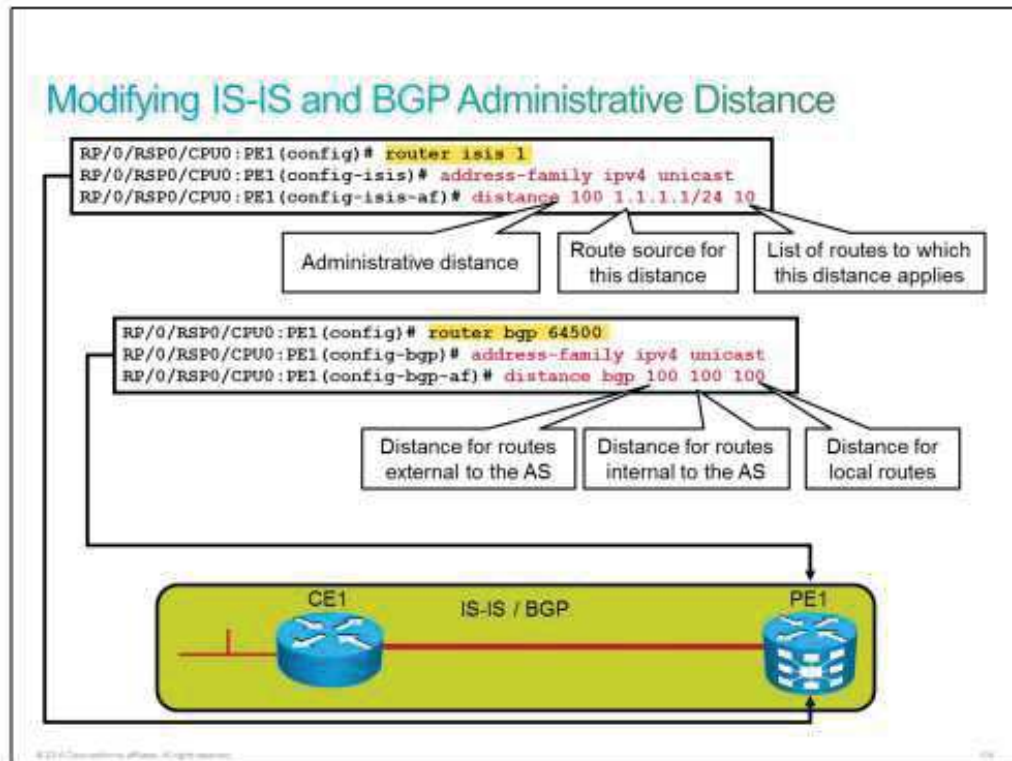
OSPF assigns different administrative distance values to external, interarea, and intra-area routes (left-side example). The default value is 110 for each of them.



OSPF allows for the modification of administrative distance only to routes that are matched with access lists or those generated by a specific OSPF source (right-side example) or both. The Cisco IOS or IOS XE or IOS XR **distance** router OSPF command modifies the administrative distance for OSPF routes.

Modifying IS-IS and BGP Administrative Distance

The administrative distance can be modified for all IS-IS routes or for individual routes from individual sources.



The Cisco IOS or IOS XE or IOS XR **distance** router IS-IS command modifies the administrative distance for IS-IS routes. In the Cisco IOS XR Software, IS-IS distance for IPv4 unicast routes needs to be applied in the IPv4 unicast address family. In the Cisco IOS or IOS XE Software, IS-IS distance for IPv4 unicast routes is applied in router IS-IS configuration mode, as shown in the example:

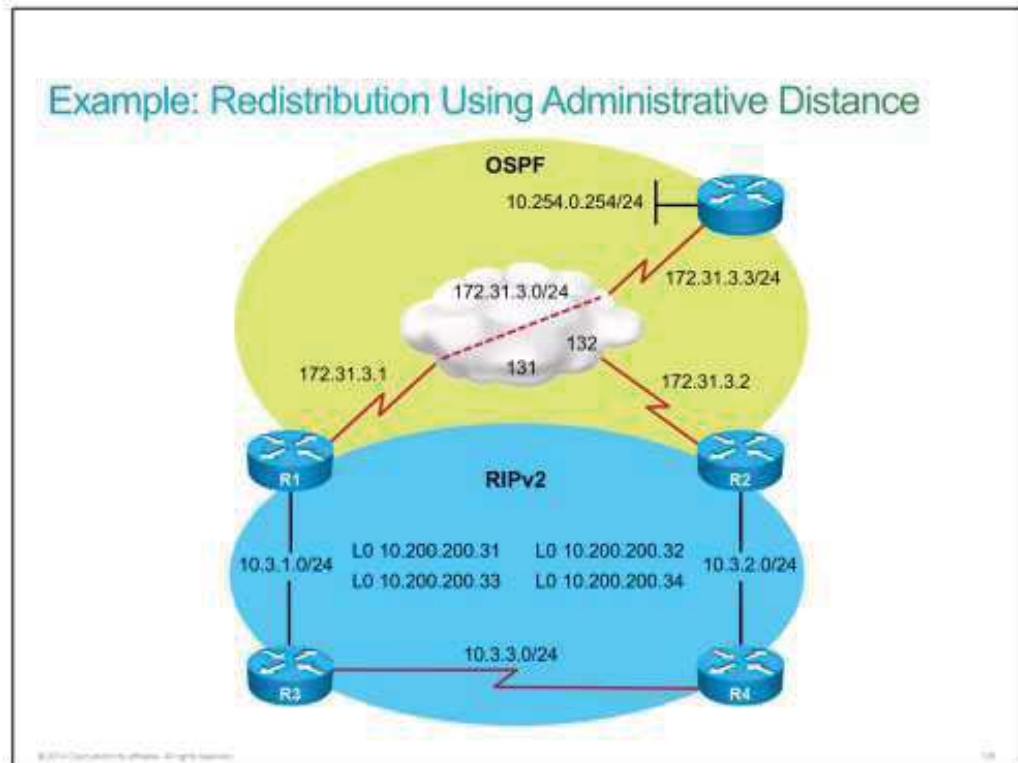
```
CE1(config)# router isis 1
CE1(config-router)# distance 100 1.1.1.1 0.0.0.0 10
```

For BGP, use the Cisco IOS or IOS XE or IOS XR **distance bgp** command. BGP assigns different administrative distance values to routes that are learned through EBGP, routes that are learned through IBGP, and local routes. The default values that are used are 20, 200, and 200, respectively. The BGP distance for IPv4 unicast routes needs to be applied in the IPv4 unicast address family. The Cisco IOS or IOS XE Software also allows you to change administrative distance only for specific BGP routes that are received from a specified BGP source, as shown in the example:

```
CE1(config)# router bgp 64501
CE1(config-router)# address-family ipv4 unicast
CE1(config-router-af)# distance 100 1.1.1.1 0.0.0.0 10
```

Example: Redistribution Using Administrative Distance

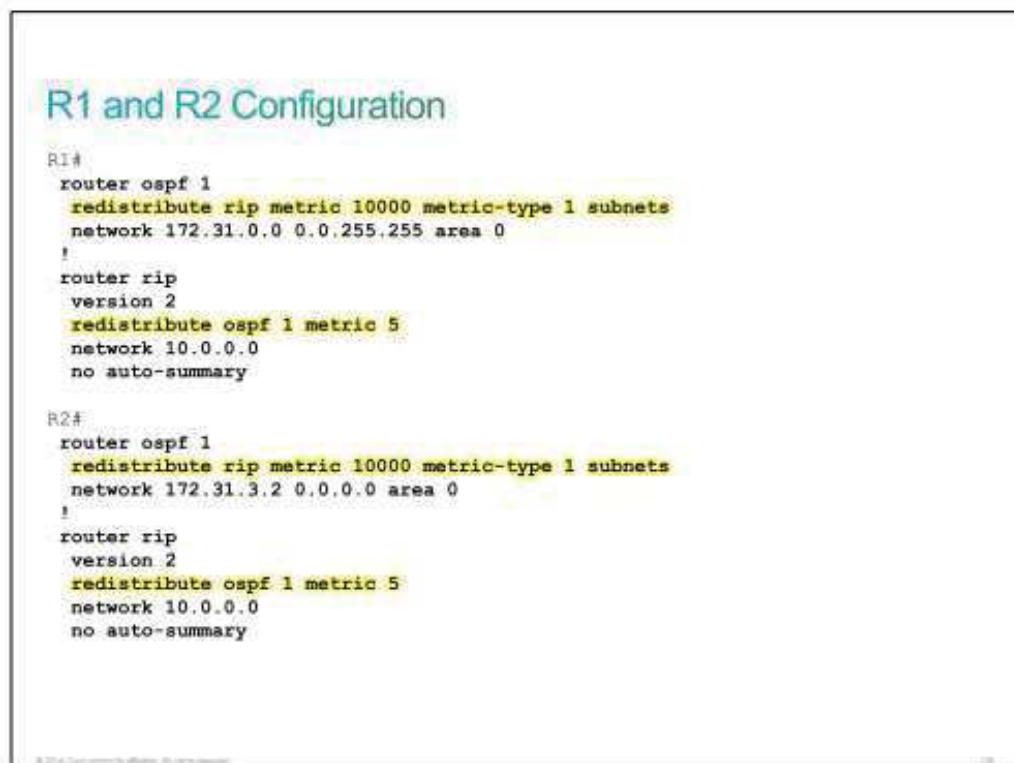
This example describes the before-and-after results of using the administrative distance to modify route redistribution.



The figure shows a network that is using multiple routing protocols. There are several ways to correct path selection problems in a redistribution environment. These examples show how problems occur, where they occur, and one possible way to resolve them. Redistribution using administrative distance is presented. The figure illustrates a network with RIP and OSPF routing domains. Recall that OSPF is more believable than RIP, because OSPF has an administrative distance of 110 and RIP has an administrative distance of 120.

If, for example, a boundary router (R1 or R2) learns about network 10.3.3.0 via RIPv2 and also via OSPF, the OSPF route will be 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 (less desirable) path.

R1 and R2 Configuration



The figure illustrates the configurations for R1 and R2. 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 to 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, also to protect against route feedback.

Suboptimal Routing on R2 After Redistribution

Suboptimal Routing on R2 After Redistribution

With OSPF and RIP running



```
R2# 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
```

- R2 includes suboptimal path and loops.

© 2014 Cisco and/or its affiliates. All rights reserved. 127

The figure displays the routing table on R2 after redistribution has occurred. R2 learned RIP and OSPF routes, but it 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 R1 first, then R2 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 R4. R4 is directly attached to R2. However, the OSPF path to that loopback interface goes through R1, then R3, and then R4 before it reaches its destination. The OSPF path that is taken is actually a longer (less desirable) path than the more direct RIP path.

Influencing Path Selection on R2

Influencing Path Selection on R2

```
R1#
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

R2#
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
```

One of the boundary routers (R2 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 shown in the figure.

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

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.

In the 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.

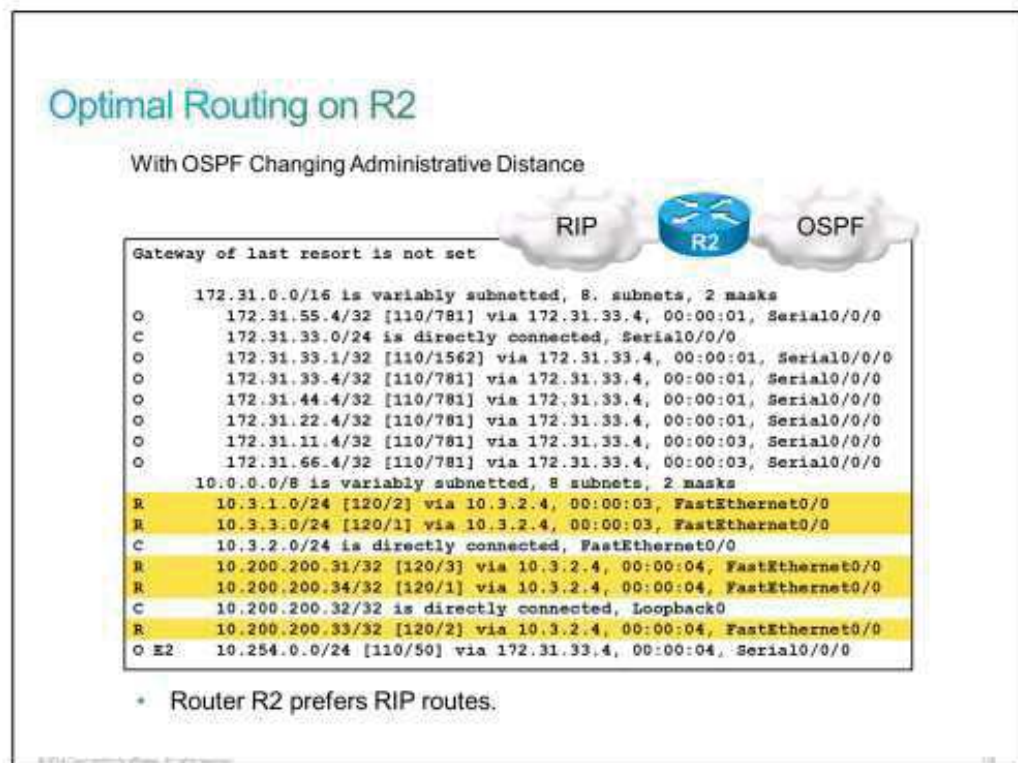
ACL 64 has permit statements for the internal native RIP networks of 10.3.1.0, 10.3.2.0, and 10.3.3.0, as well as the loopback networks of 10.200.200.31, 10.200.200.32, 10.200.200.33, and 10.200.200.34.

When either one of the redistributing routers learns about these networks from RIP, it selects the routes that are learned from RIP (with a lower administrative distance of 120) over the same routes that are 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 must configure the **distance** command on both redistributing routers, because either one of them can have suboptimal routes, depending on which redistributing router first sends the OSPF updates about the RIP networks to the other redistributing router.

Optimal Routing on R2



The figure shows that R2 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.

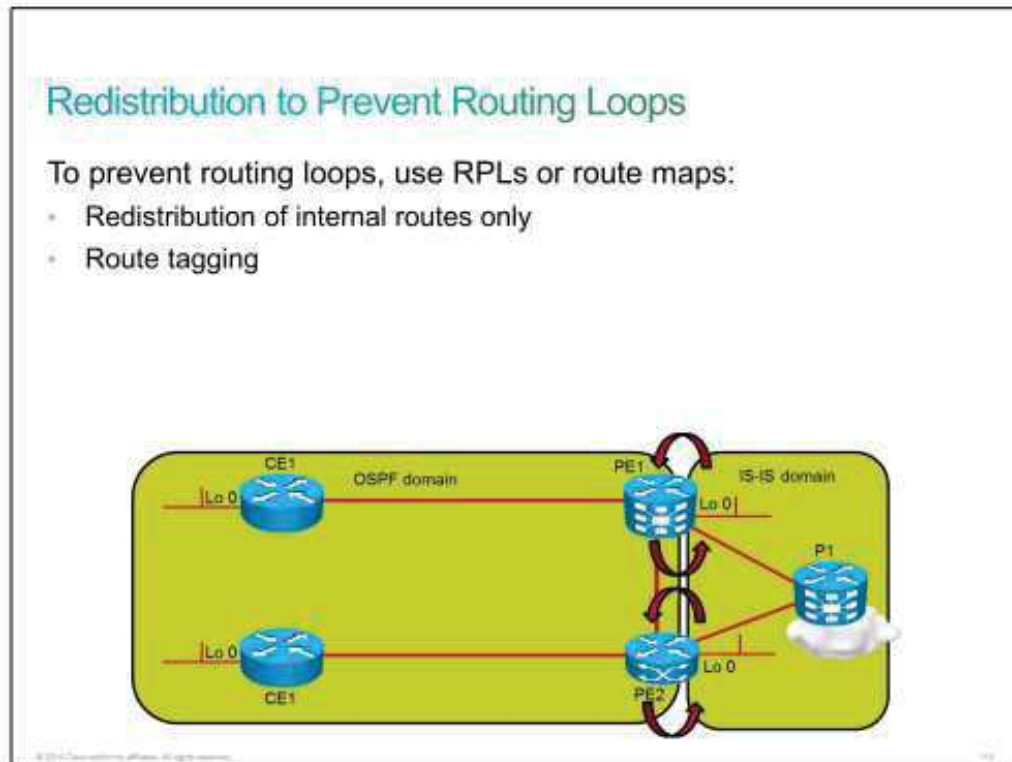
This example illustrates the importance of knowing your network before you implement redistribution, and of closely examining the results of the redistribution to determine which routes are now being selected.

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 benefit 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.

Redistribution to Prevent Routing Loops

Simple redistribution results in a loop after a route is lost.

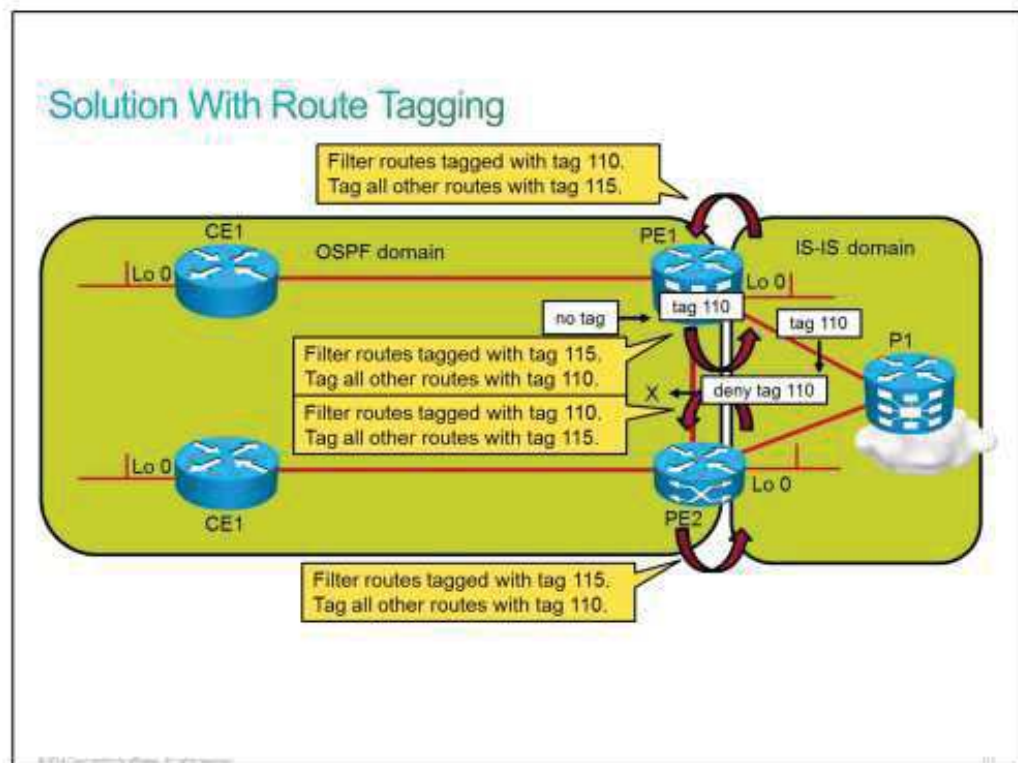


The figure represents multipoint two-way redistribution. With this kind of redistribution, it is very likely that routing loops exist. Suppose that PE1 and PE2 are both running IS-IS and OSPF routing protocols.

As networks are learned from the other IS-IS routers, they will also be redistributed into OSPF. Similarly, OSPF routes are redistributed into IS-IS. With this type of multipoint two-way redistribution, routing loops, or suboptimal routing, may occur. You can use RPLs or route maps to prevent routing loops:

- Redistributing internal routes only
- Route tagging

Solution With Route Tagging



To avoid routing loops, several solutions exist that often have to be combined. Among these solutions, you can use either RPLs or route maps to redistribute internal routes, or you can use route tagging. You can also manipulate the metric during the redistribution process.

The solution that uses RPLs or route maps and tagging is shown in the figure. IS-IS routes that are redistributed into OSPF are tagged with the value 115 during the redistribution process. In the opposite direction, routes that are redistributed into IS-IS are filtering tagged updates. Updates with a tag of 115 are not allowed to go back to the IS-IS routing protocol. This will effectively break the routing loop.

In the example, OSPF routes that are redistributed into IS-IS are tagged with the value 110 during the redistribution process. In the opposite direction, routes that are redistributed into OSPF are filtering tagged updates. Updates with a tag of 110 are not allowed to go back to the OSPF routing protocol.

Summary

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

Summary

- Redistribution is needed to exchange routing updates when having more than one routing protocol.
- The seed metric is used as initial metric of a redistributed route. By default, routing protocols have different seed metrics.
- One-point redistribution defines only one redistribution point between two routing protocols.
- Multipoint redistribution is redistribution between two routing protocols that takes place on two or more separate devices that are running both routing protocols.
- Routing loops can be introduced when you use multipoint two-way redistribution.
- Administrative distance is used to rank the trustworthiness of a routing protocol. You can manipulate it to prevent suboptimal routing.
- You can use route tagging to prevent routing loops when performing multipoint redistribution.

© 2014 Cisco and/or its affiliates. All rights reserved.

148

Lesson 4: Influencing BGP Route Selection

Overview

Routes that are learned by routers, via [BGP](#), have properties associated with them that aid a router in determining the best route to a destination when there are multiple paths to that destination. These properties are referred to as BGP attributes. This lesson introduces the role of BGP attributes, and explains how their presence influences route selection in BGP. Understanding how BGP attributes influence route selection is required for the design of robust networks.

This lesson provides advanced information on how to connect customers to service providers, and to connect service providers to upstream service providers. It includes an in-depth description of the BGP attributes that are used in route selection, including weight, local preference, [AS](#) path prepending, [MED](#), and BGP communities.

Upon completing this lesson, you will be able to:

- Describe the use of BGP weights to influence the BGP route-selection process
- Describe how to configure per-neighbor weights
- Describe how to change BGP weights using RPLs or route maps
- Describe the order of operation in setting BGP weights
- Describe how the BGP local preference attribute influences BGP route selection
- Describe how to change the local preference
- Describe how to monitor the local preference values
- Describe the function of AS path prepending and how you can use it to facilitate proper return path selection
- Describe design considerations for implementing AS path prepending
- Describe how to configure AS path prepending

- Describe how to monitor AS path prepending
- Describe how AS path prepending can impact AS path filtering
- Describe how MED can be used to facilitate proper return path selection
- Describe how to change the MED
- Describe how to monitor MED values
- Describe how BGP communities facilitate proper return path selection
- Describe how to design a BGP solution around BGP communities
- Describe how to configure BGP communities
- Describe the BGP named community lists
- Describe the use of sequenced entries in extended community lists
- Describe how to set attributes based on community values
- Describe how to monitor BGP community values
- Show examples of using BGP communities

BGP Weight

When connections to multiple providers are required, BGP must select the optimum route for traffic to use. The optimum, or best route may not be what the network designer intended, based on design criteria, administrative policies, or corporate mandate. Fortunately, BGP provides many tools for administrators to use to influence route selection. One of these tools is the weight attribute.

BGP Weight

Influencing BGP route selection: outbound traffic

- BGP routing policy can be specified by using:
 - **Weight:** provides local routing policy (within a router)
 - **Local preference:** provides AS-wide routing policy
- BGP weights are specified per neighbor.
 - Default weight
 - AS path-based weight
 - Complex criteria with RPLs or route maps

BGP route selection criteria take the weight parameter into consideration first. If a router has two alternative paths to the same destination, and their weight values are different, BGP selects the route with the highest weight value as the best. Only when the two alternatives have equal weight is the next criterion, local preference, checked.

A high local preference value is preferred over a low value. Only when the two alternatives have an equal local preference is the next criterion checked.

The weight attribute is local to a single router only. The weight value is never propagated by the BGP protocol, and this value constitutes a routing policy local to the router.

Local preference is assigned to a route as an attribute. This attribute is carried with the route on all internal BGP sessions. In this situation, all other BGP-speaking routers within the AS receive the same information. Normally, a router assigns a local preference to a route that is received on an external BGP session before it is accepted and entered in the BGP table of the border router. Routers propagate the local preference attribute on internal BGP sessions only. This policy constitutes a routing policy for the entire AS.

The router can assign the weight attribute to a route in two ways:

- All routes that are received from a specific neighbor can be assigned a default weight value. This weight value indicates that the neighbor is preferred over the other neighbors.
- A RPL command or route map that is applied on incoming routes from a neighbor can be used to select some routes and assign them weight values.

If configured, the default weight assignment on routes that are received from a neighbor is applied first. All routes that are received from the neighbor are assigned a weight value as defined by the default weight.

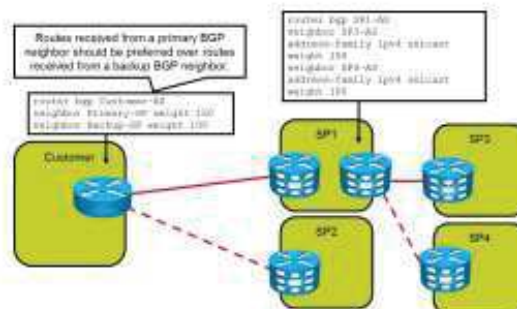
Configuring Per-Neighbor Weights

All routes that are received from the neighbor after the configuration line is in place are assigned the weight value. If no weight value is specified, the default value of 0 is applied.

Configuring Per-Neighbor Weights

BGP neighbor weights characteristics:

- All routes from the BGP neighbor get the specified weight.
- BGP routes with a higher weight are preferred.



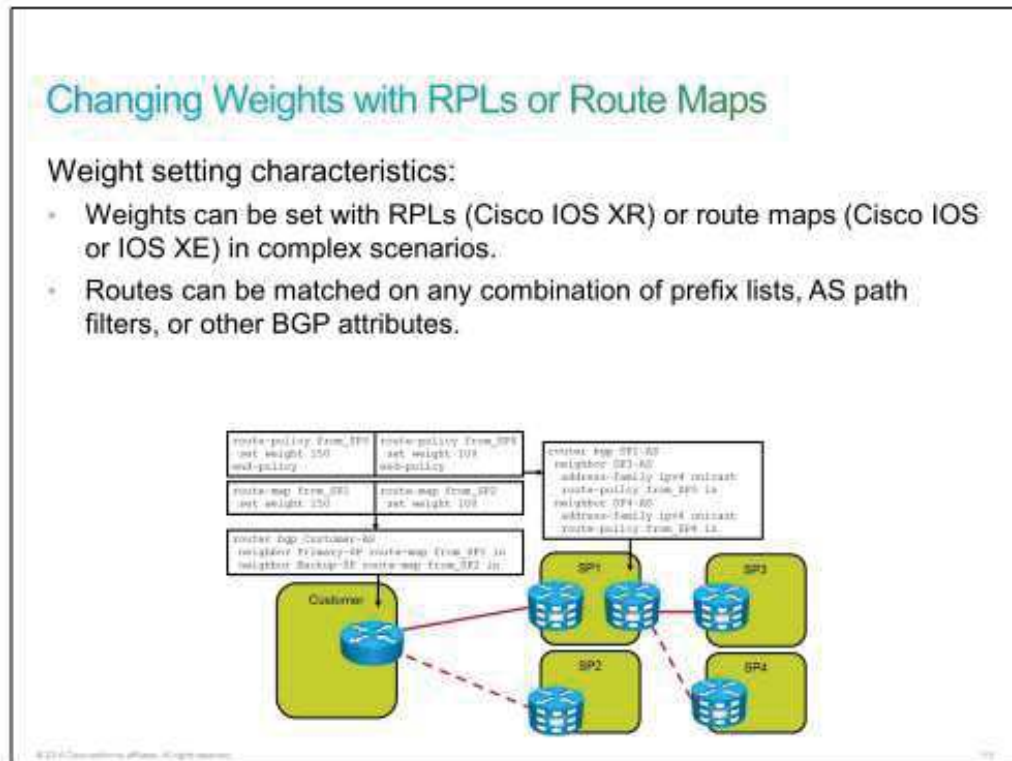
The weight is configured on both BGP sessions, giving a higher weight to the routes that are received from the primary BGP neighbor, compared to those that are received from the backup BGP neighbor.

Verify BGP weights for routes received from a neighbor by using the Cisco IOS or IOS XE **show ip bgp** or Cisco IOS XR **show bgp** commands.

Weight is applied only to new incoming updates. To enforce new weights, re-establish BGP sessions with your neighbors by using the Cisco IOS or IOS XE **clear ip bgp** or Cisco IOS XR **clear bgp** commands.

Changing Weights with RPLs or Route Maps

The RPL and route map are powerful tools to select and alter routing information. The RPL is used in the Cisco IOS XR Software while the route map is used in the Cisco IOS or IOS XE Software.



When an RPL or route map is applied to incoming information from a BGP neighbor, each received update is examined as it passes through the RPL or route map.

These examples show the RPLs and route maps used in the figure:

```

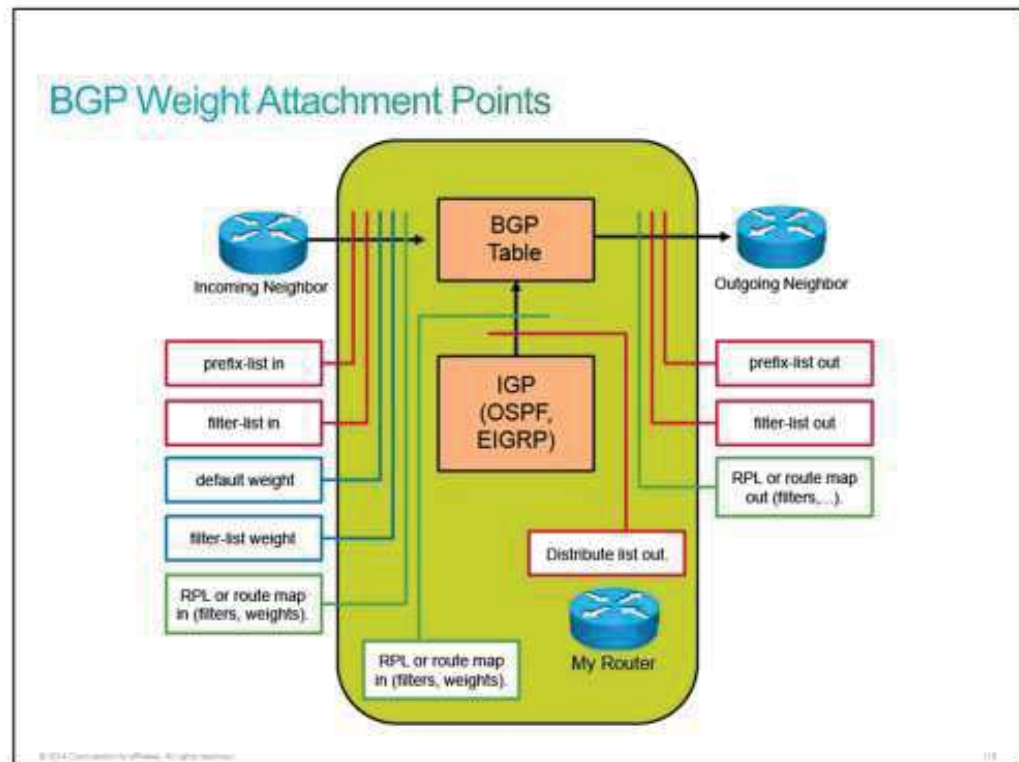
Customer#
route-map from_SP1 permit
set weight 150
!
route-map from_SP2 permit
set weight 100
!

SP1#
route-policy from_SP1
set weight 150
!
route-policy from_SP2
set weight 100
!

```

BGP Weight Attachment Points

Prefix lists and filter lists, both in and out, filter out routes and discard those that are not permitted. Weight setting is applicable only on incoming routes because a router never propagates the weight attribute to its neighbors.

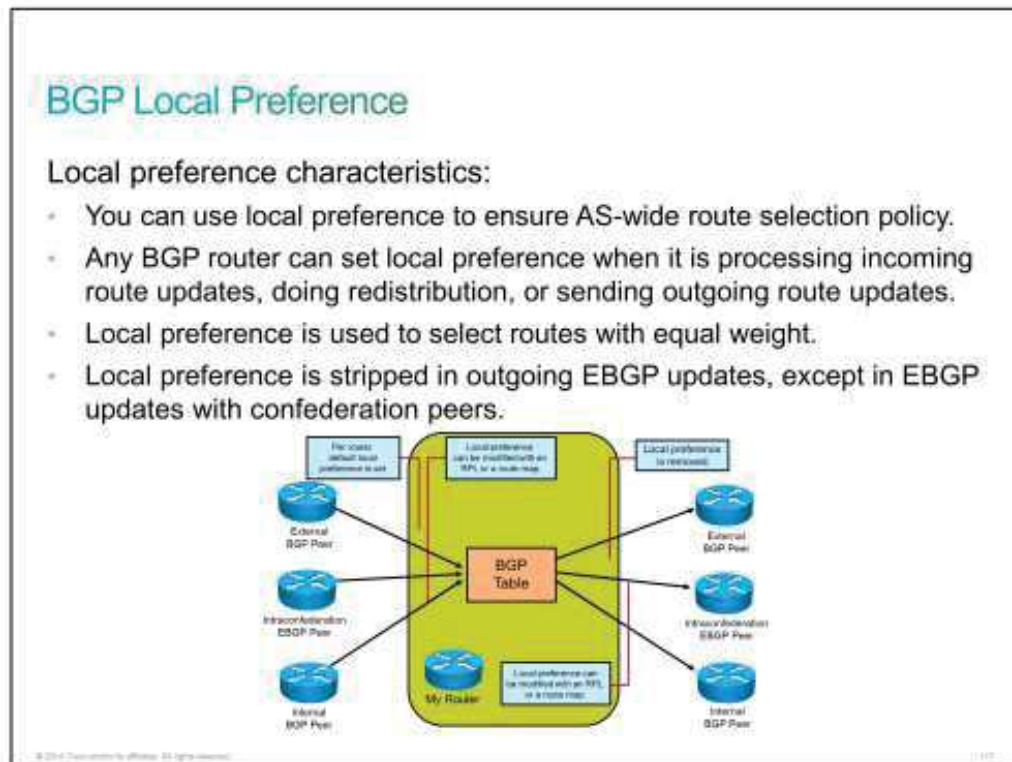


The figure shows all the possible applications of prefix lists, filter lists, weights, RPLs, and route maps. They are applied in the order indicated.

RPLs or route maps can be filters that discard routes, but they can also be used to modify and set various attributes on both incoming and outgoing routes.

BGP Local Preference

BGP route selection can be influenced by setting the BGP local preference attribute of incoming BGP routes. Local preference is similar to the weight attribute, except that the weight attribute is local to the specific router on which it is configured.



Local preference is similar to weight; because it is an attribute, you can set it once and then view it on neighboring routers without having to reset it. This attribute has a default value of 100, which the router will apply to locally originated routes and updates that come in from external neighbors. Updates that come from internal neighbors already have the local preference attribute.

Local preference is the second-highest attribute in the BGP route-selection sequence. If there are two or more paths available for the same network, a router will first compare weight, and if the weights are equal for all paths, the router will then compare the local preference attributes. The path with the highest local preference value will be preferred.

The local preference attribute is automatically stripped out of outgoing updates to EBGP sessions. This practice means that you can use this attribute only within a single AS to influence the route-selection process.

BGP Local Preference (Cont.)

Local preference influencing the route selection:

- Local preference is the second highest attribute in the BGP route selection sequence.
- Remember the BGP route selection rules:
 - Highest weight preferred (local to router)
 - Highest local preference preferred (global within AS)
 - Other BGP route selection rules
- Weights configured on a router override local preference settings.
- To ensure consistent AS-wide route selection:
 - Do not change local preference within the AS.
 - Do not use BGP weights.

© 2014 Cisco and/or its affiliates. All rights reserved.

137

Local preference is the second-highest attribute in the BGP route selection sequence. Remember the route selection rules:

1. Prefer the highest weight (local to router).
2. Prefer the highest local preference (global within AS).
3. Process all remaining BGP route selection rules.

Because network administrators can use both weight and local preference to manipulate the route selection process, they must decide which one to use. If local preference is used, the weight should be the same for all paths.

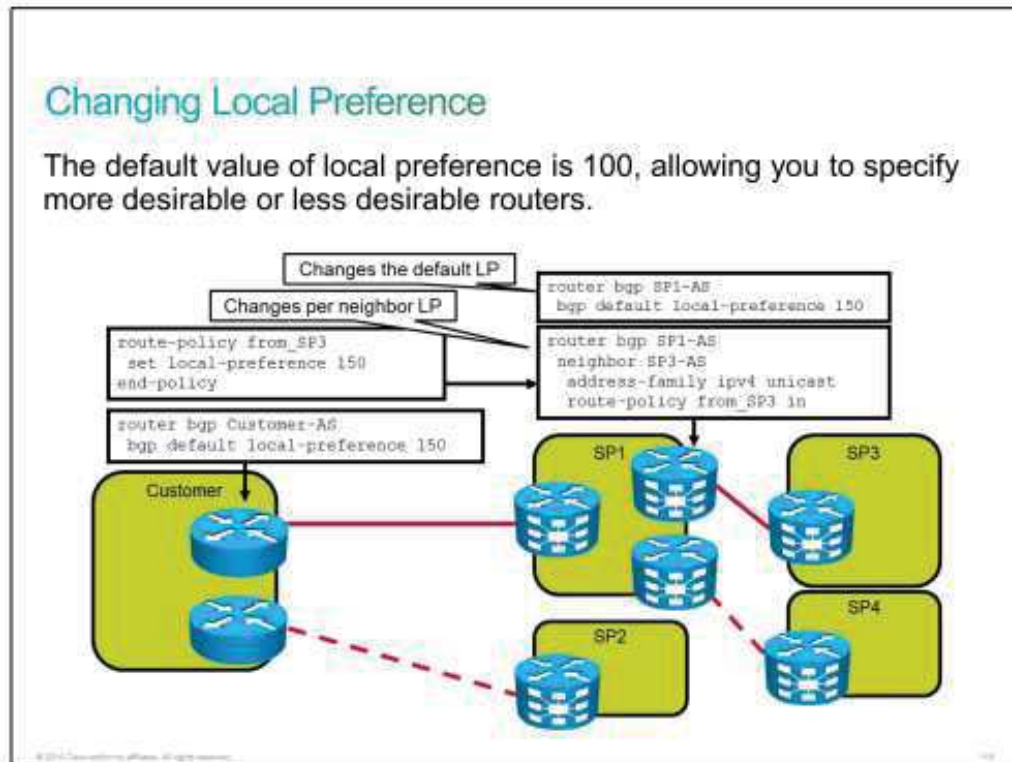
Network administrators can use weight on an individual router to override local preference settings that are used in the rest of the AS.

In most cases, it is enough to change the default local preference on updates coming from external neighbors. Network administrators should avoid changing the local preference attribute on internal sessions to prevent unnecessary complexity and unpredictable behavior.

The BGP weight is local to the router, while most commonly used attributes in the BGP route selection are BGP local preference, AS path, and [MED](#).

Changing Local Preference

A default local preference value is applied to all routes that do not have local preference set (EBGP routes).



Network administrators can apply local preference in the following ways:

- Use an RPL or route map with the Cisco IOS or IOS XE or IOS XR **set local-preference** command. You can use the RPL or route map on incoming updates from all neighbors, or on outgoing updates to internal neighbors (not recommended).
- Use the Cisco IOS or IOS XE or IOS XR **bgp default local-preference** command to change the default local preference value that is applied to all updates that come from external neighbors, or that originate locally.

Setting a value lower than the default of 100 will result in the router preferring internal paths to external paths (normally a router would prefer external routes). Setting a value higher than 100 will result in external paths being preferred to all internal paths (also those with a shorter AS path).

Monitoring Local Preference

Although local preference is not a mandatory attribute, it is applied to every route. When you are using the Cisco IOS or IOS XE **show ip bgp** or Cisco IOS XR **show bgp** commands, a locally applied default value is not shown. All other values are displayed.

Monitoring Local Preference

```
RP/0/RSP0/CPU0:PE1# show bgp
<... output omitted ...>
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop        Metric LocPrf Weight Path
*> 10.1.1.1/32     0.0.0.0          0         32768 i
*> 10.1.10.1/32    192.168.101.11   0         0 64500
*> 10.2.1.1/32     10.2.1.1         0        100
*> 10.2.10.1/32    10.2.1.1         0        100 0 64502 i
< text omitted >
```

Nondefault LP is displayed.

LP from the IBGP peer is displayed.

```
RP/0/RSP0/CPU0:PE1# show bgp 10.1.10.1/32
<... output omitted ...>
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    10.0.1.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    10.0.1.1
    64501
    192.168.101.11 from 192.168.101.11 (10.1.10.1)
      Origin IGP, metric 0, localpref 100, valid, external, best, group-best
      Received Path ID 0, Local Path ID 1, version 9
```

LP is always displayed.



© 2014 Cisco Systems, Inc. 139

You should use the Cisco IOS or IOS XE **show ip bgp prefix** or Cisco IOS XR **show bgp prefix** command to also display the locally applied value.

AS Path Prepending

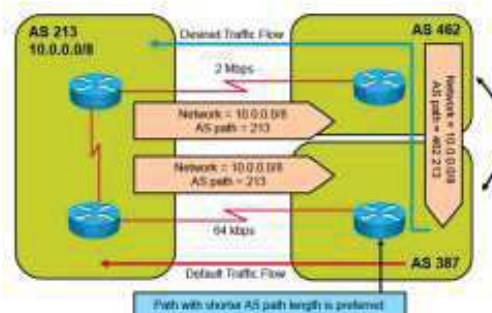
Problems can arise when administrative policies mandate that a specific return path be used for traffic that is returning to the AS, but AS path prepending potentially allows the customer to influence the route selection of its service providers.

Return Path Selection in a Multihomed AS

Return Path Selection in a Multihomed AS

Return path selection characteristics:

- **Requirement:** The return traffic to the customer must arrive over the highest-speed access link.
- **Result:** The return traffic flows over the path with the shortest AS path length.



Selecting the appropriate path for outgoing traffic is fairly easy for an AS. Influencing other autonomous systems to select the appropriate path for traffic that is returning to a specific AS is much more complicated.

Configuring the preferred path for outgoing traffic only (and not for incoming or return traffic) is likely to result in an asymmetrical traffic flow, as well as suboptimal performance of the return traffic. In the figure, outgoing traffic is directed to the high-speed line (2 Mbps) as a result of configuring local preference or weight. However, the return traffic from AS 387 would take the default path over the low-speed line (64 kbps). The low-speed line would be a limiting factor in the overall performance of that network.

In this example, AS 213 requests AS 387 to send packets toward network 10.0.0.0/8 via AS 462. The reason for this request is to improve network performance and minimize delay (assuming, of course, that the connectivity between AS 387 and AS 462 is better than the direct 64-kbps link between AS 387 and AS 213).

If no BGP path selection tools are configured on the route to influence the traffic flow, AS 387 will use the shortest AS path. This action will result in unwanted behavior, because the return traffic to AS 213 will be sent over the low-speed WAN link.

Proper Return Path Selection

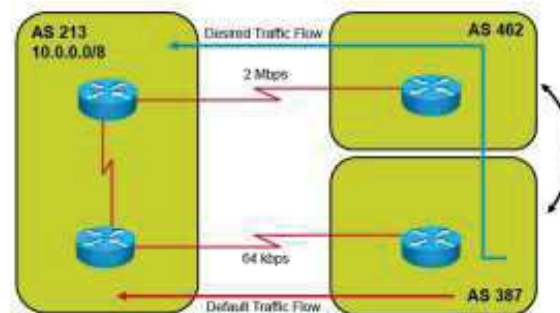
Proper Return Path Selection

Q: How do you select the proper return path from AS 387?

A: Use local preference in AS 387.

Q: Will the administrator of AS 387 configure it?

A: Unlikely.



Remember that the incoming traffic flow (from the perspective of AS 213) will be a result of the route selection for outgoing traffic in AS 387. The traffic that is going out from AS 387 will end up as incoming traffic in AS 213.

If AS 387 configures some changes that cause the route selection process for outgoing traffic to prefer to reach network 10.0.0.0/8 via AS 462, the changes would result in behavior matching the desired administrative policy for AS 213, which specifies that incoming traffic to the AS should be received over the high-speed link.

One way to accomplish the desired administrative policy in AS 213 is to configure the router in AS 387, which is receiving EBGP updates directly from AS 213, to assign a local preference value less than the default value (100) to all routes that are received from AS 213. The router in AS 387 is also configured specifically not to set local preference on EBGP routes that are received from AS 462. This configuration results in assignment of the default value of 100 to all routes received from AS 462. When the route selection process in AS 387 selects the best route to reach network 10.0.0.0/8, the difference in local preference values causes AS 387 routers to select the path via AS 462 as the best.

However, all the configuration work to complete this process must be performed in AS 387. The network administrators of AS 387 would be required to modify the router configurations in AS 387 to satisfy the administrative policy requirements of AS 213. All changes must be documented and maintained according to the rules and procedures that have been adopted by AS 387.

If AS 387 is a major ISP, the network administrators are probably too busy doing other things to tailor router configurations that are based on the demand of a single-leaf (nontransit) AS that lacks bandwidth, on a redundant connection.

AS Path Prepending

AS Path Prepending

AS path prepending characteristics:

- BGP route selection uses these criteria:
 - Prefer largest weight.
 - Prefer largest local preference.
 - Prefer routes that the router originated.
 - **Prefer shorter AS paths.**
 - Use other route-selection rules.
- Manipulating the outgoing AS path length (called **AS path prepending**) could result in proper return path selection.
- The AS path should be extended with multiple copies of the **AS number of the sender**.
- AS path prepending is used to achieve these goals:
 - Ensure proper return path selection.
 - Distribute the return traffic load for multihomed customers.

Recall that BGP route selection uses the following criteria:

- Prefer the largest weight.
- Prefer the largest local preference.
- Prefer routes that the router originated.
- Prefer shorter AS paths.
- Then, prefer all other route-selection criteria.

It is unlikely that the operator of an AS can request changes in router configurations in another AS. This limitation makes it virtually impossible to influence another AS to select the desired path, based on the weight and local preference attributes, because both options would require configuration changes in the neighboring AS.

But if both the weight and the local preference parameters are left at their default settings, they will not indicate a difference. This configuration causes the route-selection process to continue down the list of selection criteria. The third criterion for selection will not influence route selection in this scenario, because none of the routes originated at the router that is performing the route selection. The fourth criterion will apply, however, because the AS paths have different lengths.

If the AS path is not manually manipulated by some administrative means, the path going over the fewest number of autonomous systems is selected by the router regardless of available bandwidth. However, if the AS that is attempting to influence the incoming traffic flow is sending out EBGp updates with a manipulated AS path attribute over that undesired path, the receiver of this update is less likely to select it as the best because the AS path now appears to be longer.

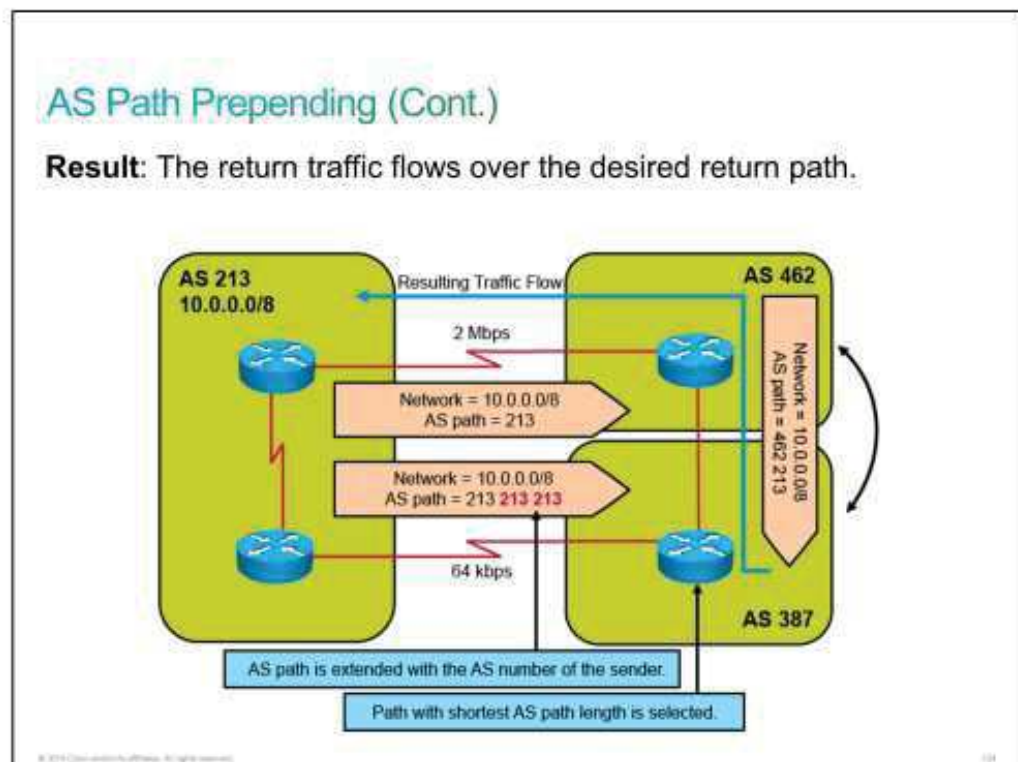
The benefit of manipulating AS paths to influence the route selection is that the configuration that is needed is done in the AS that is requesting a desired return path.

You can manipulate AS paths by prepending AS numbers to existing AS paths. Normally, you perform AS path prepending on outgoing EBGP updates over the nondesired return path. Because the AS paths sent out over the nondesired link become longer than the AS path sent out over the preferred path, the nondesired link is now less likely to be used as the return path.

The length of the AS path is extended because additional copies of the AS number of the sender are prepended to (added to the beginning of) the AS path attribute. To avoid clashes with BGP loop-prevention mechanisms, no other AS number, except that of the sending AS, should be prepended to the AS path attribute.

If another AS number is prepended in the AS path, the routers in the AS that has been prepended will reject the update because of BGP loop-prevention mechanisms.

You can configure prepending on a router for all routing updates that you send to a neighbor or only on a subset of them.



As long as the high-speed link between AS 213 and AS 462 is available, all traffic should flow toward AS 213 using the high-speed link.

To accomplish this goal, you can configure the router in AS 213 that sends EBGP updates to AS 387 by prepending the AS path with two copies of the AS number 213. AS 387 receives two alternative routes to reach network 10.0.0.0/8: the update that it has received directly from AS 213 (that has a manipulated AS path with a length of three) and the update that it has received via AS 462 (that was not manually manipulated and therefore contains an AS path length of two).

When AS 387 starts the route-selection process to determine which route to use to reach network 10.0.0.0/8, it checks the AS path length after the weight and local preference parameters. In this case, neither weight nor local preference has been configured, so the length of the AS path will be the deciding factor in the route selection process. Consequently, AS 387 prefers the shortest AS path and thus forwards packets toward network 10.0.0.0/8 via AS 462. The desired administrative policy has been met, and AS 213 will receive incoming traffic over the high-speed link.

If the forwarding path from AS 387 via AS 462 to AS 213 and network 10.0.0.0/8 is later broken, the BGP update to reach network 10.0.0.0/8 is revoked. In case of such a network failure, AS 387 will have only one remaining path to reach network 10.0.0.0/8. The route-selection process now has only one choice, the route directly to AS 213 over the low-speed WAN link. The low-speed link will therefore serve as backup to the high-speed WAN link.

AS Path Prepending Design Considerations

How many copies of the AS number of the sender should you prepend to the AS path? The answer depends on the goals of the administrative policy. In the general case, it is not easy to determine the exact number of required AS numbers to prepend. The sending AS does not know what alternative paths are available to other autonomous systems.

AS Path Prepending Design Considerations

Considerations when designing AS path prepending:

- There is no exact mechanism to calculate the required prepended AS path length.
- If a primary and backup scenario is desired, consider this strategy:
 - Use a long prepended AS path over the backup link to ensure that the primary AS path will always be shorter.
 - A long backup AS path consumes memory on every Internet router.
 - Experiment with various AS path lengths until the backup link is idle.
 - Add a few more AS numbers for additional security (unexpected changes in the Internet).
- If traffic load distribution is desired, consider this strategy:
 - Start with a short prepended AS path, monitor the link use, and extend the prepended path length as needed.
 - Continuously monitor the link use and change the prepended AS path length if required.

© 2014 Cisco and/or its affiliates. All rights reserved.

13

The following are two typical cases in which you can use AS path prepending for return path selection:

Establishing a primary link and a backup link: As an announced backup (prepended) route propagates through the Internet, all the routers along the way that receive the route need to store it together with its AS path attribute. If this information is long, it will consume extra memory in these routers. However, because routers forward only the routes that are selected as best, an AS that receives multiple alternatives to a destination will select the route with the shortest AS path and forward only that route.

If both the primary and the secondary link are up, the neighboring AS will receive two routes to the same destination that differ only in the AS path length. The route with the shorter AS path will be subsequently advertised through the Internet.

If the primary link fails, the route with the longer AS path is the only remaining route. As a result, the primary route is withdrawn, and the prepended route is advertised through the Internet. In this case, extra memory will be consumed in each Internet router because of the storage of the prepended (longer) AS path.

The longer the AS path that is announced to the [EBGP](#) neighbor on the other side of the backup link, the less likely it is that incoming traffic will be received from that neighbor. The network administrator can make a clever guess about how many copies of the AS number to prepend. After the prepending is implemented, the network administrator has to examine the result. If the expected result is not achieved, the configuration can be changed and a few more copies of the AS number can be prepended.

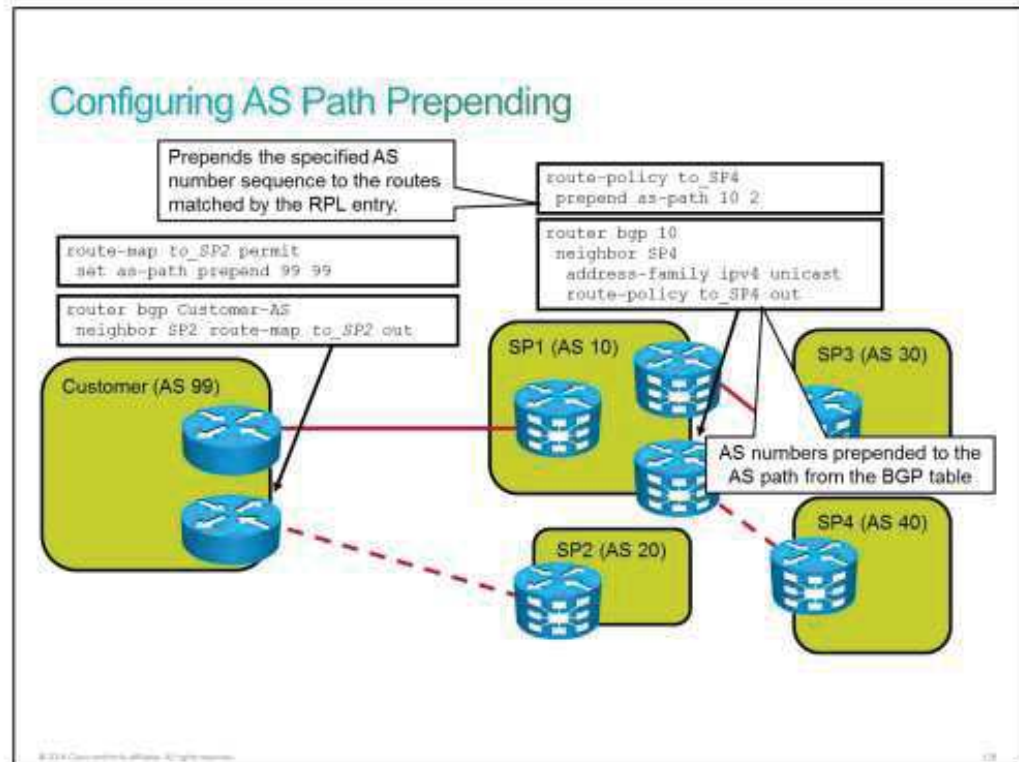
After AS path prepending has generated the desired results, the network administrator may take the precaution of prepending a few more copies of the AS number to the AS path. This action protects the customer from packets being routed over the backup link at a possible later stage when the topology between remote autonomous systems has unexpectedly changed, yielding a longer AS path to reach the primary link.

Distributing the load of return traffic: In a multihomed scenario, there is no way to predetermine the exact volume of traffic that will be received over a particular link. The traffic load on different links will change, depending on where the senders are located (in which autonomous systems they belong). The network topology and the way that different remote autonomous systems are interconnected may also change with time, changing the load distribution. Only constant monitoring and fine-tuning will ensure that the desired results are achieved.

In a first attempt at load distribution, the network administrator can configure a router that is connected to an overused link to prepend only a few extra copies of the local AS number. After the network has been given time to converge, the network administrator must check the change in load distribution. The monitoring of the load must be done for a period long enough to be statistically significant (several days or more). If enough volume of traffic has not moved from the overused link to the underused link, the administrator must prepend more copies of the local AS number, and the process of resending local routes and monitoring the results starts all over again.

Configuring AS Path Prepending

You can configure manual manipulation of the AS path attribute (prepending) using an RPL or route map.



To configure manual manipulation of the AS path attribute (prepending) use the Cisco IOS or IOS XE route map **set as-path prepend** command or the Cisco IOS XR RPL **prepend as-path** command. The RPL or route map is used to prepend the specified AS numbers to outgoing EBGP route updates that are matched with the **match** condition. AS path prepending is completed first, and then the route is subject to the normal AS path modification procedures when it is sent over an EBGP session.

Monitoring AS Path Prepending

When you are monitoring AS path prepending, the router doing the prepending is not the proper point to observe the results of the AS path prepend operation.

Monitoring AS Path Prepending

AS path prepending monitoring options:

- AS path prepending cannot be monitored or debugged on the sending router.
 - **debug bgp (debug ip bgp) update** displays the BGP entry prior to RPL or route-map processing.
 - **show policy-map (show route-map)** does not display how many routes have matched a RPL or route-map entry.
- The results of AS path prepending can be observed on the receiving router.

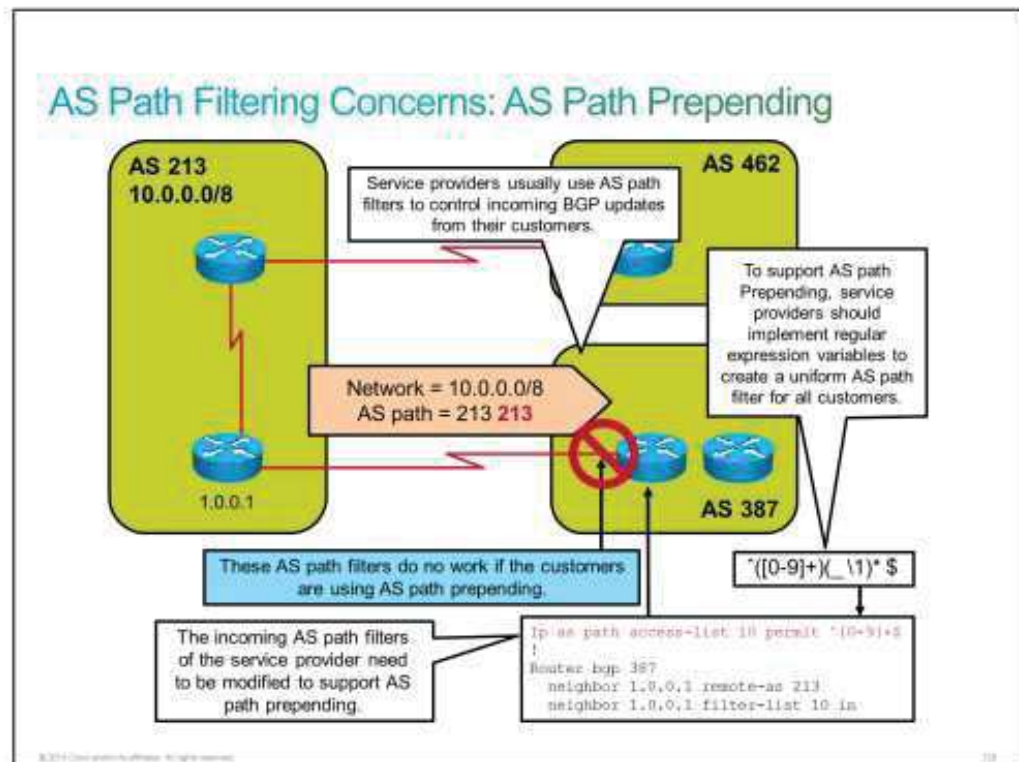
Output from the Cisco IOS or IOS XE **debug ip bgp updates** or the Cisco IOS XR **debug bgp update** command does not display the prepended paths, because the RPL or route map doing the prepending is applied afterward.

The Cisco IOS or IOS XE **show route-map** or the Cisco IOS XR **show policy-map** command displays the configuration details of an RPL or route map. The matching criteria and AS path manipulation are displayed as output of the command. However, there is no indication of how many routes have been matched by an RPL or route map statement, and have therefore had their AS paths manipulated.

A better place for observing AS path prepending is on the router that is receiving the BGP update that contains the prepended AS path. At that point, you can use the pattern of AS number sequences in the received AS path attribute of received routes to find the routes that have a prepended AS path.

AS Path Filtering Concerns: AS Path Prepending

Service providers normally expect their customers to send routes that originate only in the AS of the customer. However, because customers might not do so, proactive thinking and care for the rest of the Internet cause the service provider to implement AS path filters on incoming updates that are received from their customers.



The network administrator of the service provider in the figure could configure individual filters for each neighbor. However, a single AS path access list permitting only AS paths with a length of exactly one AS number would be a better solution because the service provider can uniformly apply it to all incoming routes from all customers.

In the figure, the service provider (AS 387) has configured a filter list, which allows only AS paths that have a length of one AS number. When the customer changes its router configuration and starts to announce network 10.0.0.0/8 with a prepended AS path, the filter-list for incoming routes to AS 387 in the service-provider router will filter those routes out. This filtering results in a situation where the network 10.0.0.0/8 is not reachable over the link between AS 213 and AS 387. Therefore, the backup function is not available.

Network 10.0.0.0/8 is, however, still reachable via the path going through AS 462. This situation means that AS 387 can send packets to network 10.0.0.0/8 but not over the direct link to AS 213. This failure may be difficult to detect because, during normal conditions, all autonomous systems in the figure can exchange traffic.

After AS 387 loses the route to network 10.0.0.0/8 via AS 462, possibly because the primary link between AS 213 and AS 462 is gone, the problem will be obvious. AS 387 can now no longer reach network 10.0.0.0/8 at all, although the physical link between AS 213 and AS 387 is available. Because the AS of the service provider will receive customer routes with prepended AS paths that have a length greater than one AS number, the provider must modify its incoming filters.

The service provider needs to create a new inbound regular expression filter, using regular expression variables and parentheses for recall.

What is needed is a filter that will allow any AS path containing one or multiple copies of the same AS number. An example of such a filter is as follows:

```
^([0-9]+)(_\1)*$
```

This filter matches any AS path beginning with any AS number and continues with no or multiple repetitions of that same AS number (the variable “\1” repeats the value in the brackets). The regular expression would therefore match AS paths 99 99 99, 2 2 2, or 100, but it would not match AS path 100 99.

BGP Multi-Exit Discriminators

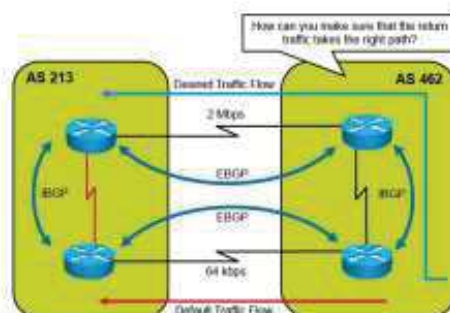
BGP provides a tool for administrators to influence route selection, the MED attribute. You can influence BGP route selection by setting the BGP MED attribute of outgoing BGP routes. Two methods that are used to set the MED attribute are the default MED and RPLs or route maps.

Selecting the Proper Return Path

Selecting the Proper Return Path

MED and the return path:

- You can use the MED to influence path selection in a neighbor AS.
- An AS can specify its preferred entry point by using the MED in outgoing EBGP updates.



When multiple connections between providers are required, BGP attributes such as weight and local preference solve only half the problem: how to choose the right path out of the AS. Here the focus will be on the second, more complex half of the problem: how to influence neighboring autonomous systems to choose the correct return path back into the AS.

The MED attribute is a hint to external neighbors about the preferred path into an AS when multiple entry points exist. You can apply the MED attribute on outgoing updates to a neighboring AS to influence the route selection process in that AS. The MED attribute is useful only when there are multiple entry points into an AS.

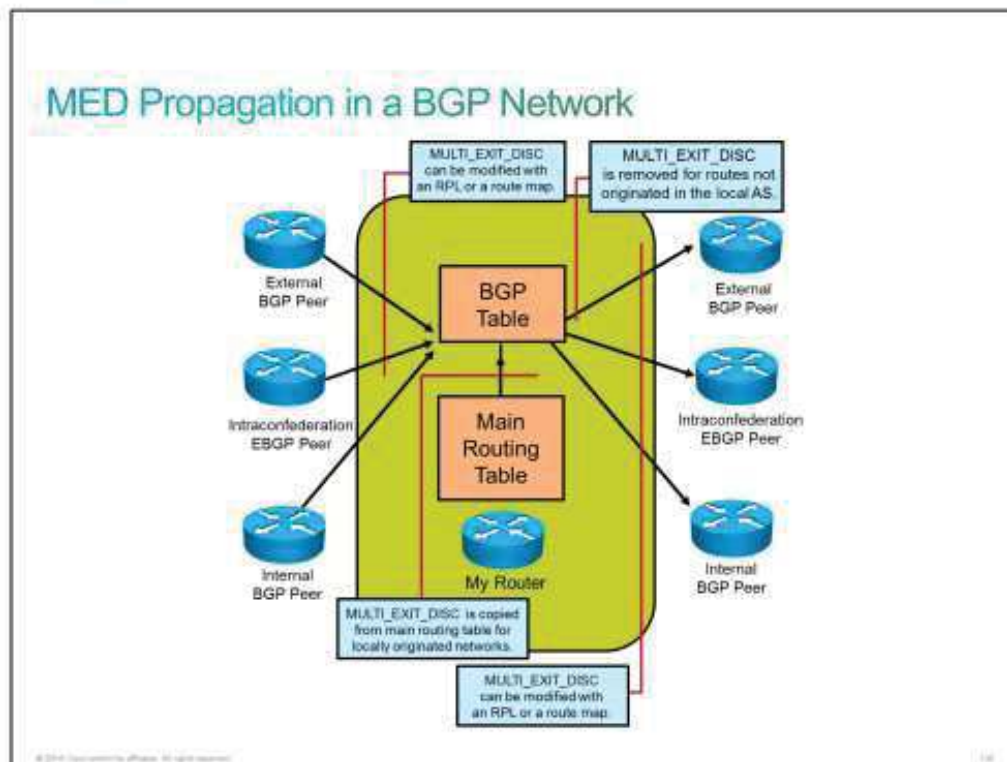
The MED attribute, which is sent to an external neighbor, will be seen only within that AS. An AS that receives a route that contains the MED attribute will not advertise that MED beyond its local AS.

The default value of the MED attribute is 0. A lower value of the MED attribute indicates a more preferred path.

The MED attribute is considered a “weak” metric. In contrast with weight and local preference, a router will prefer a path with the smallest MED value, but only if the weight, local preference, AS path, and origin code attributes are equal. Using the MED may not yield the expected result if the neighboring AS modifies any of the stronger BGP route selection mechanisms.

In Cisco IOS or IOS XE or IOS XR Software, metric is the term that is used for MED; this also applies to the Cisco IOS or IOS XE Software **set** command that is used in route maps, and in all **show** and **debug** commands. In the Cisco IOS XR Software, the term med is used in the RPL, but in all **show** and **debug** commands the term metric is used.

MED Propagation in a BGP Network



The figure shows how the value of the MED attribute is assigned, depending upon the routing information source. An RPL or route map must be configured on a router to manually assign a value to the MED attribute. For the networks that are also present in the BGP table, the router assigns a default value from the metric in the routing table and copies it into the MED attribute. The MED attribute is automatically removed on external sessions if the attribute did not originate in the local AS.

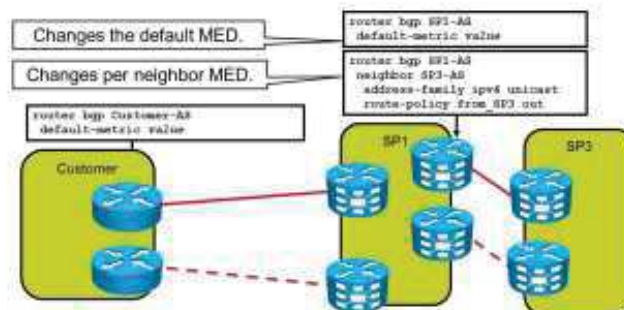
Changing the MED

The MED is not a mandatory attribute, and there is no MED attribute that is attached to a route by default.

Changing the MED

MED information:

- The MED is copied from the IGP cost in the router that sources the route (via the **network** command or through route redistribution).
- You can change the MED value for redistributed routes with the **default-metric** command.



The only exception is if the router is originating networks that have an exact match in the routing table (through the Cisco IOS or IOS XE or IOS XR **network** command or through redistribution). If that is the case, the router uses the metric in the routing table as the MED attribute value.

Using the Cisco IOS or IOS XE or IOS XR **default-metric** command in BGP configuration mode causes all redistributed networks to have the specified MED value.

You can use an RPL or route map to set the MED on incoming or outgoing updates. Use the Cisco IOS or IOS XE **set metric** command within route map or Cisco IOS XR **set med** command within RPL configuration mode to set the MED attribute.

Advanced MED Configuration

Advanced MED Configuration		
	Cisco IOS/IOS XE	Cisco IOS XR
By default, the MED is considered only during the selection of routes from the same AS. The MED is also considered for routes coming from a different AS.	bgp always-compare-med	bgp bestpath med always
If the MED is not attached to a BGP route, it is interpreted as value 0, and thus as the best metric. A missing MED is interpreted as infinity (worst).	bgp bestpath med missing-med-worst	bgp bestpath med missing-as-worst
By default, the MED is considered only during the selection of routes from the same AS, which does not include intraconfederation autonomous systems. Allow routers to compare paths learned from confederation peers.	bgp bestpath med confed	bgp bestpath med confed
Changes the BGP route selection procedure to a deterministic but slower one.	bgp deterministic-med	<i>default</i>

There are several rules about when and how you should use the MED attribute:

- You should use the MED in the route selection process only if both (all) paths come from the same AS. Use the Cisco IOS or IOS XE **bgp always-compare-med** or Cisco IOS XR **bgp bestpath med always** command to force the router to compare the MED even if the paths come from different autonomous systems. You need to enable this option in the entire AS; otherwise, routing loops can occur.
- According to a BGP standard that describes MED, you should regard a missing MED attribute as an infinite value. Cisco IOS or IOS XE or IOS XR Software, however, regard a missing MED attribute as having a value of 0. Use the Cisco IOS or IOS XE **bgp bestpath med missing-med-worst** or the Cisco IOS XR **bgp bestpath med missing-as-worst** command when combining equipment from different vendors. An even better solution is to make sure that every update carries a MED attribute.
- You must use the Cisco IOS or IOS XE or IOS XR **bgp bestpath med confed** command when you use the MED within a confederation to influence the route selection process. A router will compare MED values for the routes that originate in the confederation.
- When you enable a deterministic MED comparison, you allow a router to compare MED values before it considers BGP route type (external or internal) and IGP metric to the next-hop address. The router will compare MED values immediately after the AS path length.

Monitoring the MED

All BGP-related **show** and **debug** commands display the value of the MED attribute. If the inbound soft reconfiguration feature is enabled on the router, the original MED attribute that is received by the router is also displayed.

Monitoring the MED

```
RP/0/RSP0/CPU0:PE1# show bgp
<... output omitted ...>
Origin codes: i - IGP, e - EGP, ? - incomplete
Network      Next Hop      Metric LocPrf Weight Path
*> 10.1.1.1/32  0.0.0.0        0         32768 i
*> 10.1.10.1/32 192.168.101.11 0          0 64501 i
*> 10.2.1.1/32  10.2.1.1       0         100 0 i
*> 10.2.10.1/32 10.2.1.1       0         100 0 64502 i
< text omitted >
```

MED is displayed as metric.

```
RP/0/RSP0/CPU0:PE1# show bgp 10.1.10.1/32
<... output omitted ...>
Paths: (1 available, best #1)
  Advertised to peers (in unique update groups):
    10.0.1.1
  Path #1: Received by speaker 0
  Advertised to peers (in unique update groups):
    10.0.1.1
    64501
    192.168.101.11 from 192.168.101.11 (10.1.10.1)
    Origin IGP, metric 0, localpref 100, valid, external, best, group-best
    Received Path ID 0, Local Path ID 1, version 9
```

MED is displayed only for those routes that contain a MED attribute.



Both the original and the modified routes are displayed when inbound soft reconfiguration is configured.

To see the original MED, you need to enable soft reconfiguration on the router. The Cisco IOS or IOS XE **show ip bgp neighbor address received-routes** or the Cisco IOS XR **show bgp neighbor address received routes** command displays the original updates before any filters or RPL or route maps have filtered or changed them.

If soft reconfiguration is enabled, the original updates to the MED attribute are available by using the Cisco IOS or IOS XE **show ip bgp prefix** or the Cisco IOS XR **show bgp prefix** command. The original versions are marked with the **received-only** keyword and follow the version that is in the global BGP table.

BGP Communities

You can influence BGP route selection by setting the BGP community attribute on outgoing BGP routes.

BGP Communities

BGP communities:

- BGP communities are a means of tagging routes to ensure a consistent filtering or route-selection policy.
- The community attribute is a transitive optional attribute. Its value is a 32-bit number (range 0 to 4,294,967,200).
- The standards define several filtering-oriented communities:
 - **no-advertise**: Do not advertise routes to any peer.
 - **no-export**: Do not advertise routes to real EBGp peers.
 - **local-as**: Do not advertise routes to any EBGp peers.
 - **internet**: Advertise this route to the Internet community.
- A 32-bit community value is split into two parts:
 - High-order 16 bits contain the AS number of the AS that defines the community meaning.
 - Low-order 16 bits have local significance.

BGP communities are attributes that are used to group and filter routes. Communities are designed to give the network operator the ability to apply policies to large numbers of routes by using match and set clauses in the configuration of RPLs or route maps. Community lists are used in this process to identify and filter routes by their common attributes.

A community is an attribute that is used to tag BGP routes. A router can apply it to any BGP route by using an RPL or route map. Other routers can then perform any action, based on the tag (community) that is attached to the route. There can be more than one BGP community that is attached to a single route, but the routers, by default, remove communities in outgoing BGP updates.

The community attribute is a 32-bit transitive optional BGP attribute that is designed to group destinations and apply routing decisions (accept, prefer, redistribute, and so on) according to communities, to allow the easy application of administrative policies. BGP communities provide a mechanism to reduce BGP configuration complexity on a router that is controlling the distribution of routing information.

A set of community values has been predefined. When a router receives a route that has been marked with a predefined community, the router will perform a specific, predefined action that is based on that community setting:

- **no-advertise**: If a router receives an update carrying this community, it will not forward that update to any neighbor.
- **no-export**: If a router receives an update carrying this community, it will not propagate that update to any external neighbors, except intraconfederation external neighbors. This is the most widely used predefined community attribute.

- **local-as:** This community has a similar meaning to **no-export**, but it keeps a route within the local AS (or member AS within the confederation). The route is not sent to external BGP neighbors or to intraconfederation external neighbors.
- **internet:** Advertise this route to the Internet community. All routers belong to it.

Routers that do not support the community attribute will pass the attribute to other neighbors because it is a transitive attribute.

Community attributes are usually used between neighboring autonomous systems. For the BGP communities to be globally unique, a public AS number should be part of the community value. For this reason, you can enter the community value as two 16-bit numbers that are separated by a colon. The first number (high-order 16 bits) should be the AS number of the AS that defines the community value, and the second number should be a value that is assigned a certain meaning (that is, translation of a community value into local preference in the neighboring AS).

Communities can also be used internally, within an AS (to ensure AS-wide routing policy); in this case, the first 16 bits should contain the AS number of the local AS.

Using Communities

Using Communities

Using communities, step-by-step:

- Define administrative policy goals:
 - Solve asymmetrical customer-routing problems.
- Design filters and route selection policy to achieve administrative goals:
 - Set local preference of customer routes to 50 for customers using the backup service provider.
- Define communities that signal individual goals:
 - Community 387:17 is used to indicate that the local preference of the route should be lowered to 50.
- Configure route tagging on entry points, or let BGP neighbors tag the routes.
- Configure community distribution.
- Configure route filters and route selection parameters, based on communities.

© 2014 Cisco and/or its affiliates. All rights reserved.

157

Steps for Designing a BGP solution around BGP communities

Designing a BGP solution around BGP communities generally requires the following steps.

Steps for Designing a BGP solution around BGP communities

BGP solution around BGP communities:

1. Define the administrative policy goals.
2. Define the filters and route-selection policy that will achieve the required goals.
3. Assign a community value to each goal.
4. Apply communities.
5. Enable community distribution.
6. Match communities with route maps and route filters, change BGP attributes, or influence the route-selection process.

1. Define the administrative policy goals that you need to implement.
2. Define the filters and route-selection policy that will achieve the required goals.
3. Assign a community value to each goal. This table lists the goals and the community values. All customers of the service provider should know this list so that they can use the BGP communities without having to discuss their use with the service provider.

Goal	Community Value
Set local preference of 50.	387:17
Set local preference of 150.	387:18
Prepend the AS path once when sending the network to external neighbors.	387:21
Prepend the AS path twice when sending the network to external neighbors.	387:22
Prepend the AS path three times when sending the network to external neighbors.	387:23

4. Apply communities on incoming updates from neighboring autonomous systems or tell the neighbors to set the communities themselves.

5. Enable community distribution throughout your AS to allow community propagation.
6. Match communities with route maps and route filters, change BGP attributes, or influence the route-selection process, based on the communities that are attached to the BGP routes.

Configuring BGP Communities

When configuring BGP communities, you have to follow the procedure listed here.

Configuring BGP Communities

Use these procedures to configure BGP communities:

- Configure route tagging with BGP communities.
- Configure BGP community propagation.
- Define BGP community access lists (community lists) to match BGP communities.
- Configure RPLs or route maps that match on community lists and filter routes, or set other BGP attributes.
- Apply RPLs or route maps to incoming or outgoing updates.

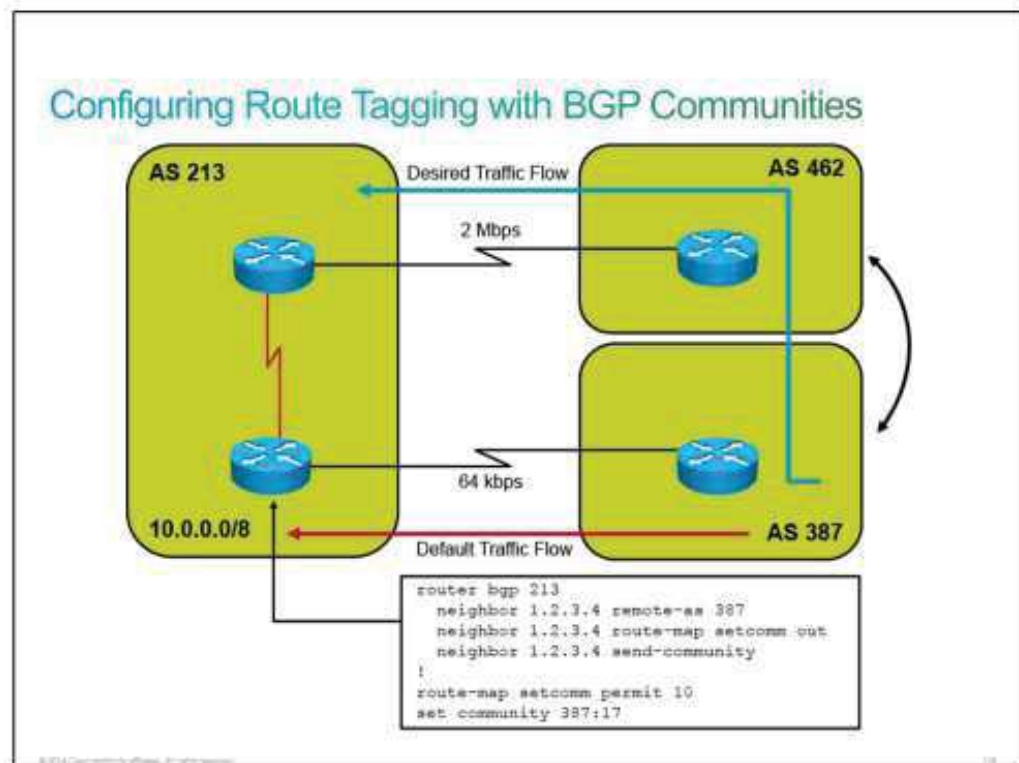
© 2014 Cisco and/or its affiliates. All rights reserved.

107

Use these procedures to configure BGP communities:

- Set communities, which requires an RPL or route map.
- Enable community propagation per neighbor for all internal neighbors. If communities are sent to external neighbors, you must enable community propagation for external neighbors.
- Create community lists to be used within RPLs or route maps to match on community values.
- Create RPLs or route maps where community lists are used, to match on community values. You can then use RPLs or route maps to filter, based on community values, or to set other parameters or attributes (for example, local preference, [MED](#), or [AS](#) path prepending).
- Apply RPLs or route maps to incoming or outgoing updates.

Configuring Route Tagging with BGP Communities

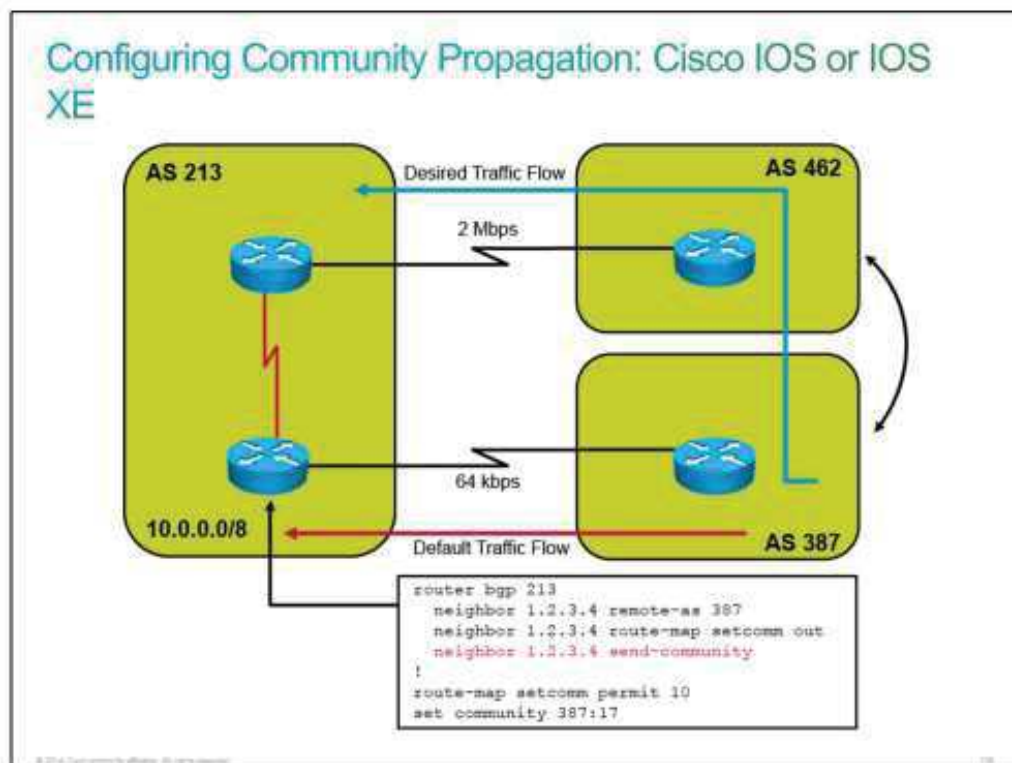


In an RPL or route map configuration mode, you should use the Cisco IOS or IOS XE or IOS XR **set community** command to attach a community attribute (or a set of communities) to a route. If the keyword **additive** is used, the original communities are preserved and the router simply appends the new communities to the route. Omitting the **additive** keyword results in the overwriting of any original community attributes.

You can apply an RPL or route map to incoming or outgoing updates. You can also use it with redistribution from another routing protocol.

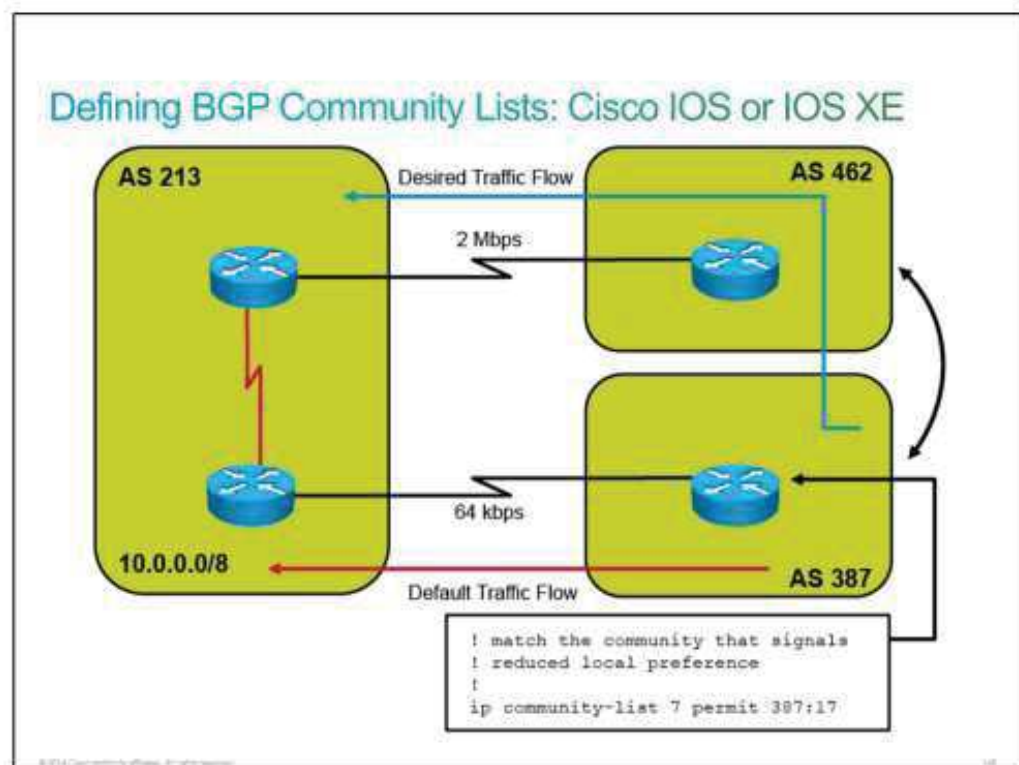
In this example, a border router in AS 213 applies a community value of 387:17 to all networks that are sent to neighboring AS 387.

Configuring Community Propagation: Cisco IOS or IOS XE



A command that is commonly forgotten by network administrators when configuring BGP communities is the Cisco IOS or IOS XE **neighbor ip-address send-community** command. This command is needed to propagate community attributes to BGP neighbors. Even if you use an outgoing route map to set communities, by default, the router will strip out any community values that are attached to outgoing BGP updates if you have not configured this command for the specific BGP neighbor.

Defining BGP Community Lists: Cisco IOS or IOS XE



You can use a standard (Cisco IOS or IOS XE) community access list to find community attributes in routing updates. A standard community list is defined by its assigned list number, which can range from 1 to 99 (Cisco IOS or IOS XE).

Community lists are similar to standard IP access lists in the following ways:

- The router evaluates the lines in the community list sequentially.
- If no line matches communities that are attached to a BGP route, the route is implicitly denied.

Standard community lists are different from standard IP access lists in the following ways:

- The keyword **internet** should be used to permit any community value.
- If more values are listed in a single line, they all have to be in an update to produce a match.

An expanded (Cisco IOS or IOS XE) community list is defined by its assigned list number, which can range from 100 to 199. Regular expressions are used to match community attributes. When a router processes a list of communities that are attached to a network update, they are converted into an ordered string of characters.

The access list has been configured to match communities that were previously set by the router in AS 213.

BGP Named Community Lists

This topic describes BGP named community lists.

BGP Named Community Lists

BGP named community list characteristics:

- Naming allows the network operator to assign meaningful names to community lists, and increases the number of community lists that can be configured.
- Named community lists can be configured with regular expressions and with numbered community lists.
- There is no limitation on the number of community attributes that can be configured for a named community list.
- The number of community lists that can be configured by a network operator increases, because there is no limitation on the number of named community lists that can be configured.

The BGP named community lists feature (Cisco IOS or IOS XE or IOS XR) introduces a new type of community list, the named community list. A named community list can be configured with regular expressions and with numbered community lists. The BGP named community lists feature allows the network operator to assign meaningful names to community lists. All the rules of numbered communities apply to named community lists except that there is no limitation on the number of community attributes that can be configured for a named community list. Although both standard and expanded community lists (Cisco IOS or IOS XE) have a limitation of 100 community groups that can be configured within each type of list, a named community list does not have this limitation.

BGP Support for Sequenced Entries in Extended Community Lists

The BGP Support for Sequenced Entries in Extended Community Lists feature allows automatic sequencing of individual entries in BGP extended community lists.

BGP Support for Sequenced Entries in Extended Community Lists

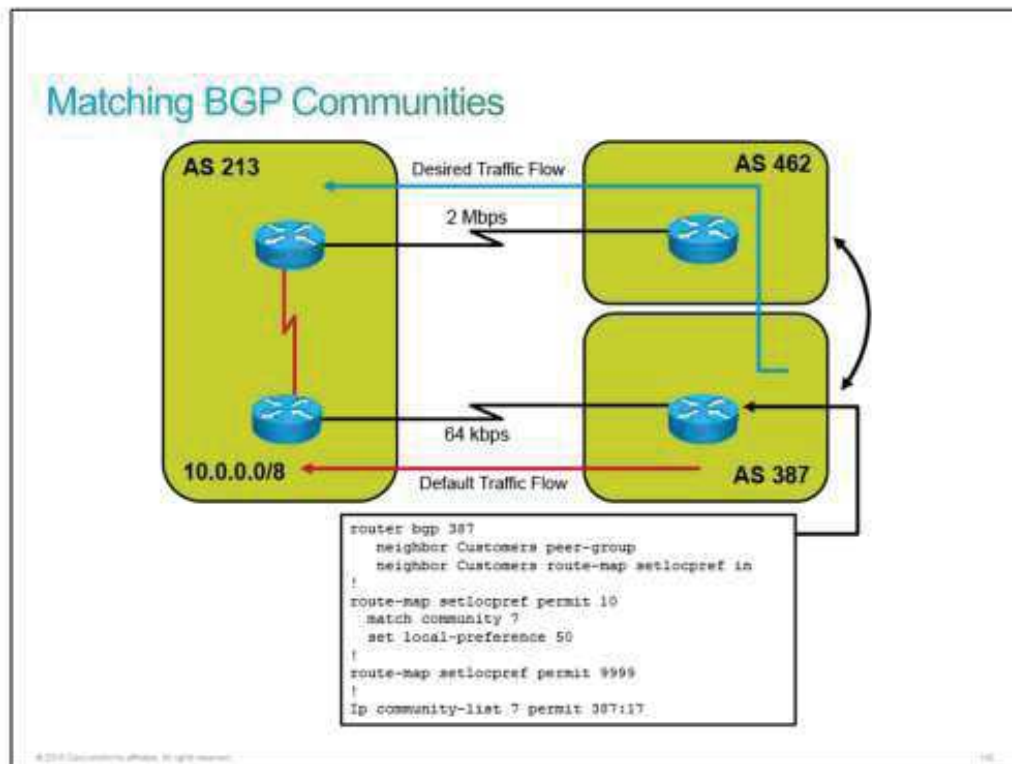
BGP Support for Sequenced Entries in Extended Community Lists characteristics:

- Allows automatic sequencing of individual entries in BGP extended community lists
- Provides the ability to remove or resequence extended community list entries without deleting the entire existing extended community list
- Configures sequence numbers for extended community list entries
- Resequences the existing sequence numbers for extended community list entries
- Configures an extended community list to use default values

This feature also provides the ability to remove or resequence extended community list entries without deleting the entire existing extended community list. Both named and numbered extended community lists can be configured in IP extended community list configuration mode.

Matching BGP Communities

Network administrators use RPLs or route maps to match networks that carry a subset of communities that are permitted by the community list. Other parameters or attributes can then be set, based on community values. If you use the keyword **exact**, all communities that are attached to a BGP route have to be matched by the community list.



You can use an RPL or route map to filter or modify BGP routing updates. Any BGP-related **set** commands can be used to set BGP parameters and attributes (that is, weight, local preference, and MED).

In the example, all updates that are received from the neighboring AS 213 are processed by the route map, which uses a community list to find community 387:17. If the community list matches one of the community attributes, the **set** command is executed and the route is permitted. If the route does not contain the right community, the route is simply permitted by the route map statement 9999 without changing anything in the update. The result is that AS 387 prefers other paths to AS 213 because they have a default local preference of 100.

Monitoring BGP Communities

Because a community is an attribute that can appear more than once in a single update, the Cisco IOS or IOS XE **show ip bgp** or the Cisco IOS XR **show bgp** command does not show it.

Monitoring BGP Communities

Monitoring communities:

- Communities are displayed in a **show bgp prefix** printout.
- Communities are not displayed in debugging outputs.
- Routes in the BGP table that are tagged with a set of communities, or routes matching a community list, can be displayed.

Monitoring BGP Communities (Cont.)

```
RP/0/RSP0/CP00:PE5# show bgp 10.5.10.1/32
< text omitted >
Paths: (1 available, best #1)
  Advertised to update-groups (with more than one peer):
    0,1
  Path #1: Received by speaker 0
  Advertised to update-groups (with more than one peer):
    0,1
64505
  192.168.105.51 from 192.168.105.51 (10.5.100.1)
    Origin IGP, metric 0, localpref 100, valid, external, best, group-best
    Received Path ID 0, Local Path ID 1, version 66
  Community: 1:100
```

```
RP/0/RSP0/CP00:PE5# show bgp route-policy com1
< text omitted >
      Network          Next Hop          Metric LocPrf Weight Path
*> 10.5.10.1/32        192.168.105.51      0           0  64505  i
* 10.5.100.1/32       192.168.105.51      0           0  64505  i
```

You can view communities only if you use the Cisco IOS or IOS XE **show ip bgp prefix** or the Cisco IOS XR **show bgp prefix** command.

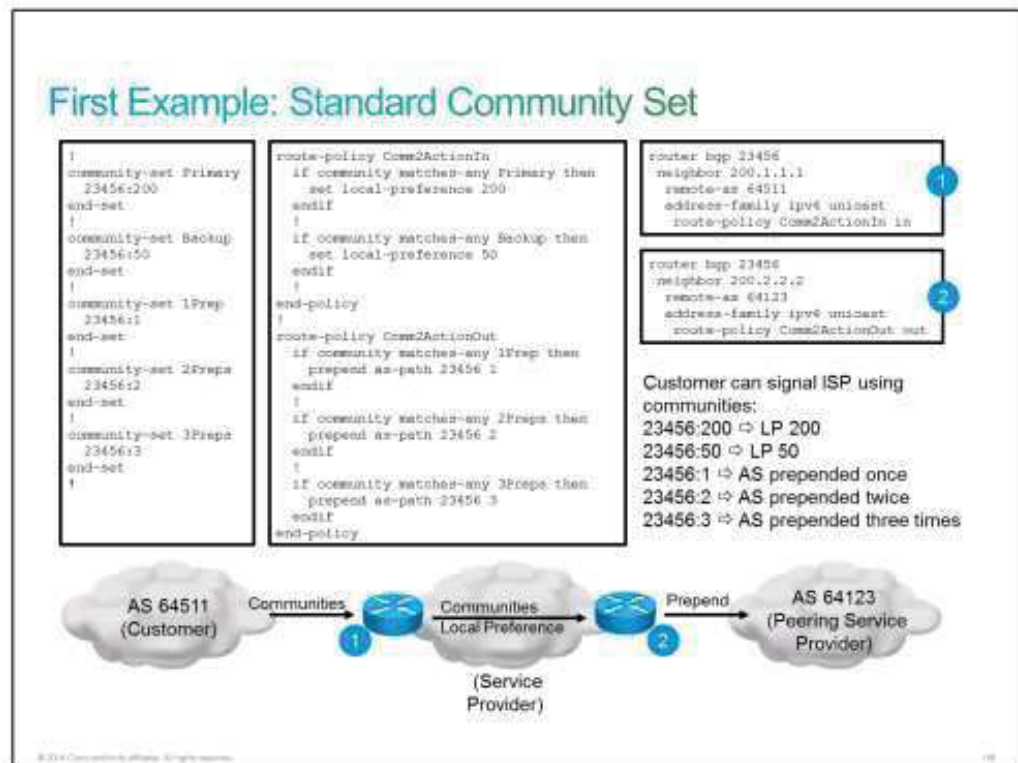
If you use the Cisco IOS or IOS XE **show ip bgp community-list** command, all networks that are permitted by the community list are listed. In the Cisco IOS XR, you have to attach the community list to the RPL as shown in the following example, to display BGP routes with specified communities:

```
community-set 1
  1:100
end-set
!
route-policy com1
  if community matches-any 1 then
    pass
  endif
end-policy
```

BGP Community Examples

Examples of using BGP communities are shown here.

First Example: Standard Community Set

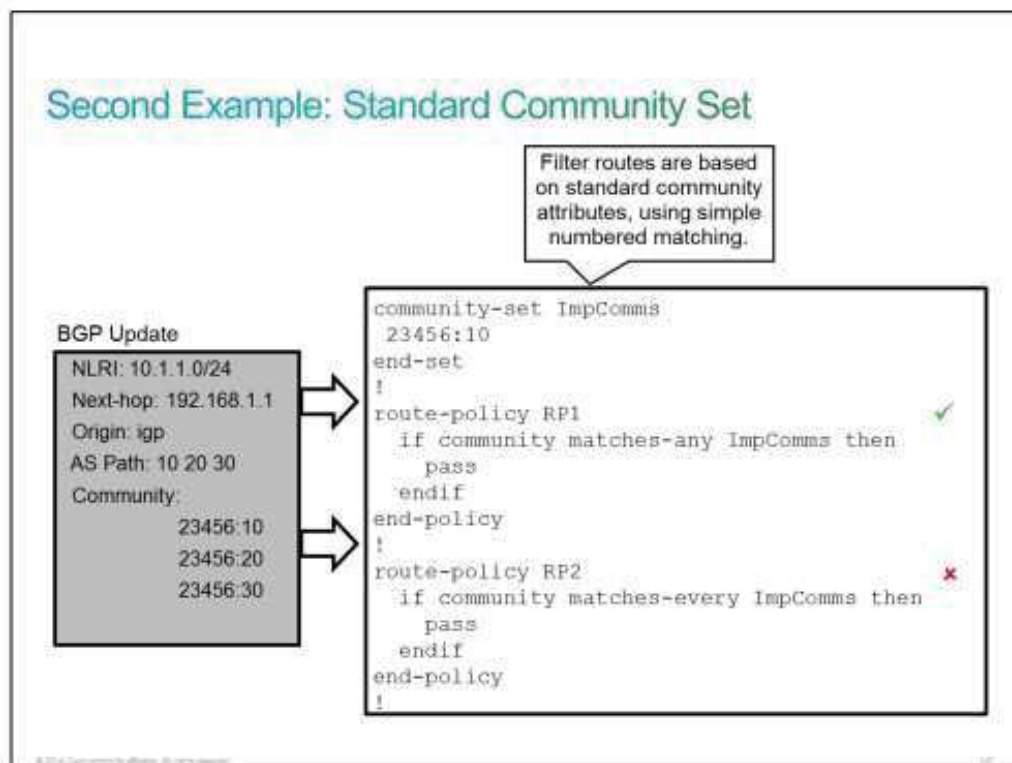


The sample configuration illustrates an AS-wide implementation of a policy:

- Allow customers to signal preference using BGP communities, that are translated into appropriate local preference values
- Additionally, the egress routers perform prepending on behalf of customers if they have tagged the routes with appropriate BGP communities.

In this example, five community sets are used to match on BGP community attributes that are coming from external neighbors (that is, a customer). A customer can, for example, signal that he wants to use this service provider for a backup connection and this customer may choose to attach two BGP communities to achieve the desired goal, 23456:50 and 23456:3. The route policy **Comm2ActionIn**, used in the inbound direction on AS edges, will apply some action, based on the matched communities. The second "if" statement will match the first community and set the local preference to 50, thus making it less desirable than some other paths that will have the default local preference, 100. The **Comm2ActionOut**, used in the outbound direction on AS edges, will match the second community in the third "if" statement and prepend the AS path attribute three times, using its own AS number.

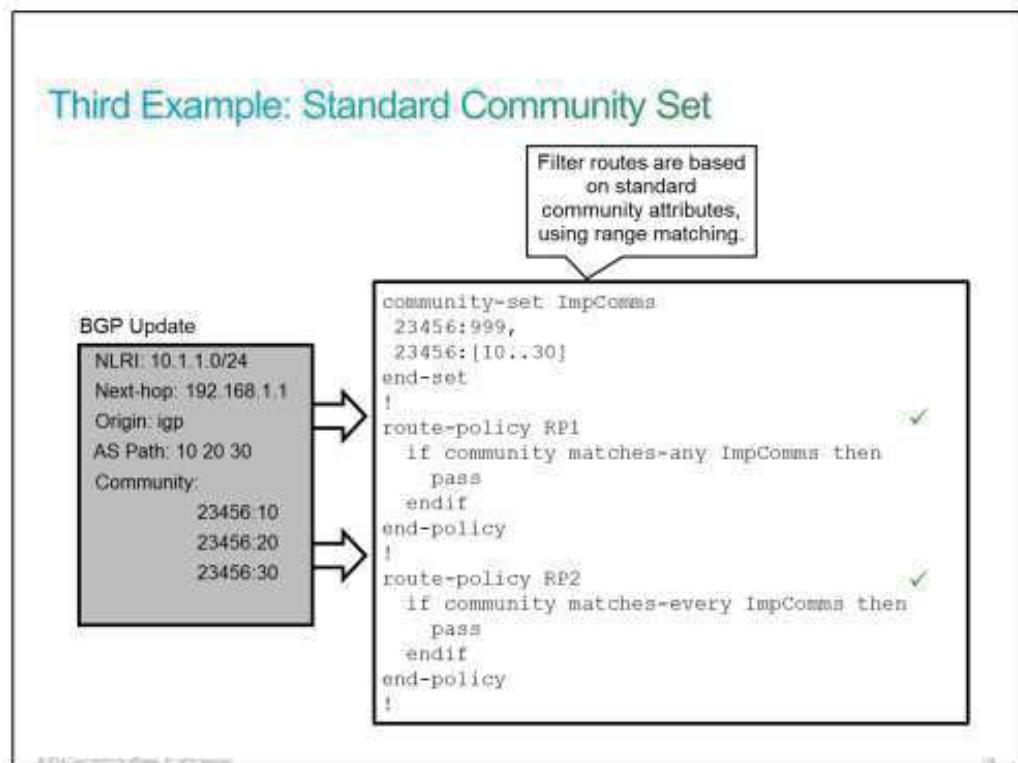
Second Example: Standard Community Set



The second sample configuration illustrates the difference between the **community matches-any** command and the **community matches-every** command:

- RP1: the route with the three community values will match the community set ImpComms, because it contains the 23456:10 community.
- RP2: the route with the three community values will not match the community set ImpComms because it does not match for two community values (23456:20 and 23456:30)

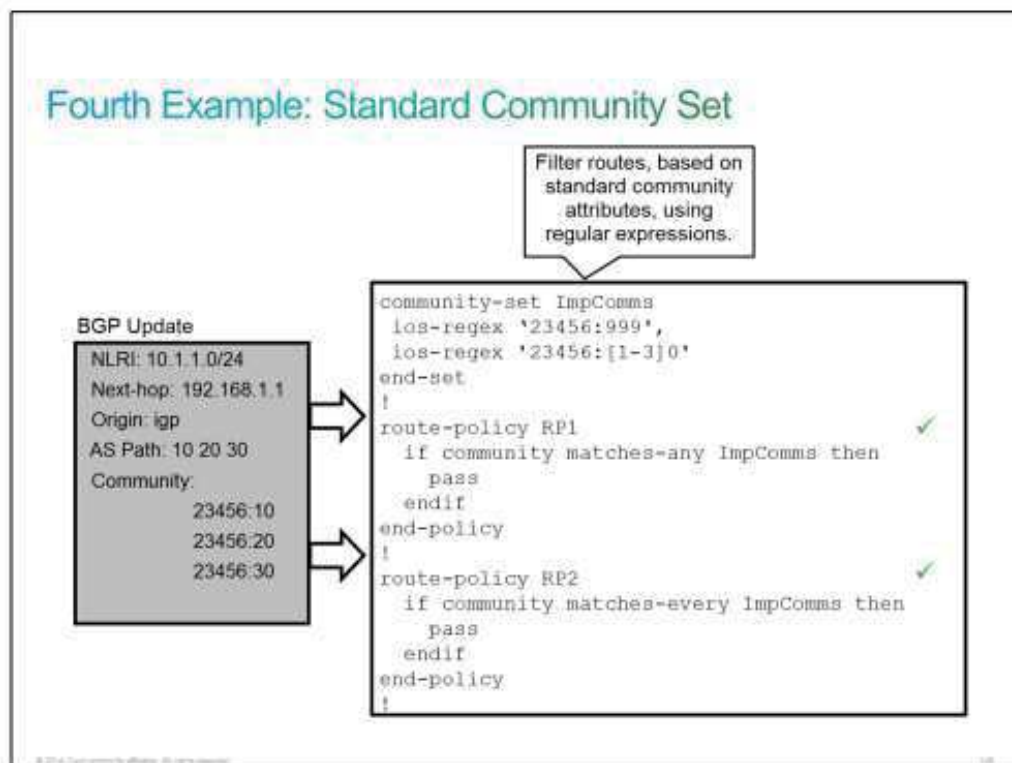
Third Example: Standard Community Set



The third example shows that the route with three community values will match in both policies, because all three community values are matched by the modified community set, ImpComms, that now contains range-based matching for 23456:10-23456:30.

The community set in the example uses number-based matching.

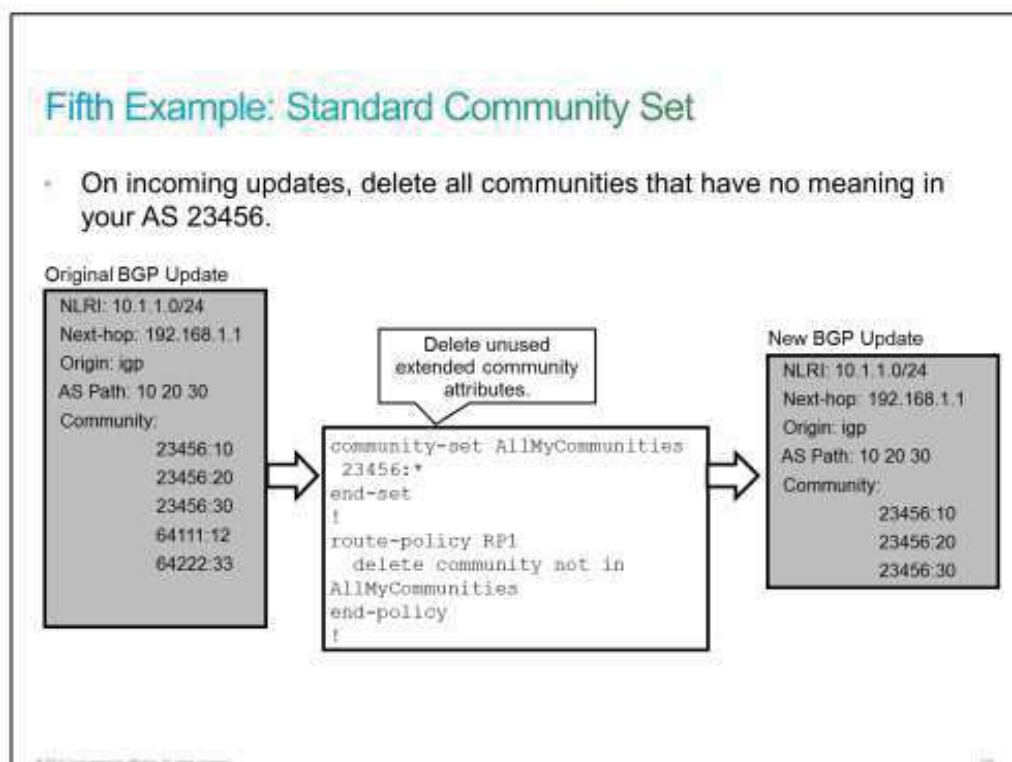
Fourth Example: Standard Community Set



The fourth example shows the same result as the previous one, except that this example uses regular expression-based matching.

The regular expression `23456:[1-3]0` will match `23456:10`, `23456:20`, and `23456:30`.

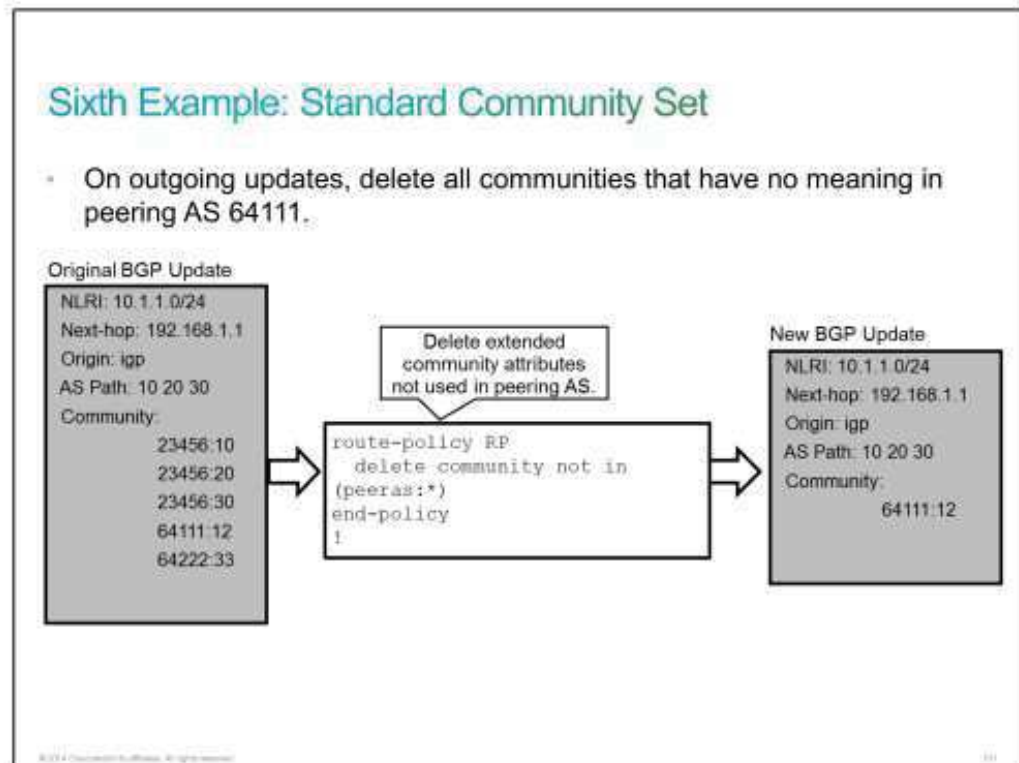
Fifth Example: Standard Community Set



The configuration in the fifth example shows how to delete communities from incoming updates that are outside the desired range (only keep communities that have meaning in the local AS).

This would be a common filter that a service provider would use to strip the updates of any BGP communities that have no meaning in its AS. The numbered matching specifies the AS number 23456 of the service provider, and matches any community value for this AS, using the wildcard ("*"). The route policy then deletes all but those communities that are in this range.

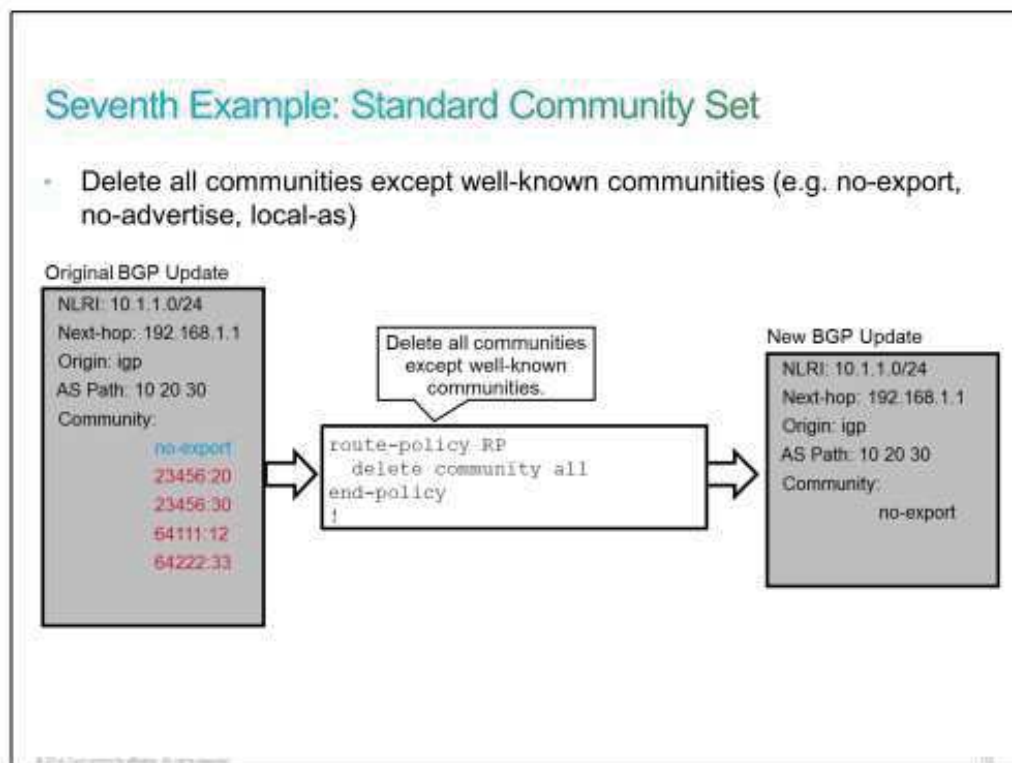
Sixth Example: Standard Community Set



The configuration in example six shows how to delete, from outgoing updates, the communities that are outside the desired range (only keep communities that have meaning in the neighboring AS).

Like the previous example, a service provider can strip out any BGP communities that have no meaning in the neighboring AS. The built-in **peer-as** keyword can be used to automatically match the neighbor AS number, and the wildcard, to match any subsequent value. Instead of using a named community set, the example uses an inline community set defined within parentheses.

Seventh Example: Standard Community Set



The configuration in example seven shows how to delete all communities, by using the **all** keyword in place of the community set.

Note This command does not remove the well-known communities (for example, **no-export**) which have a predefined action and must be explicitly deleted if required.

As shown in this example, all communities (except the well-known community **no-export**) have been removed from the update.

Summary

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

Summary

- BGP weights can be used to influence the BGP route-selection process.
- Weight can be configured on a BGP session and applied to incoming BGP updates.
- Local preference is similar to the weight attribute in that, you can use both to influence BGP path selection, but it differs from the BGP weight attribute in that weight is local to the specific router on which it is configured.
- To influence incoming path selection, you can use AS path prepending which is performed on outgoing EBGP updates over the nondesired return path, or the path where the traffic load should be reduced.
- Service providers should take into account possible AS path prepending that is done by customers when designing AS path filters.

Summary (Cont.)

- The MED is a "weak" parameter in the route selection process; it is used only if weight, local preference, AS path, and origin code are equal. By default, the MED is compared only for paths that were received from the same AS.
- BGP communities are a means of tagging routes to ensure consistent filtering or routing policy.
- You can use the BGP community attribute to create an AS-wide routing policy or to provide services to neighboring autonomous systems.
- The BGP named community lists feature allows the network operator to assign meaningful names to community lists.
- You can use BGP communities to allow customers to signal preference of a specific path to the service provider. The service provider then performs route manipulation based on received communities.

Lesson 5: Module Summary

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

This module identified mechanisms that are available, in combination with routing protocols, to filter routing information or to implement desired routing policies in service provider networks. Also covered in this module were IP route redistribution and the control of redistributed routing updates. The use of attributes to influence BGP route selection was discussed and was illustrated with examples.

Module Summary

- Tools for routing protocol manipulation include prefix-based filtering for security and AS path-based filtering for flexibility.
- You should use RPLs or route maps to implement modular and complex routing policies.
- Routes are redistributed into a routing protocol by using the **redistribute** command under the routing process. Extreme care has to be taken, when you are configuring multipoint redistribution, to prevent suboptimal routing and routing loops.
- Weight, local preference, AS path prepending, MED, and BGP communities can be used to manipulate BGP path selection and enforce routing policies.

© 2014 Cisco and/or its affiliates. All rights reserved.

177

Lesson 6: 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.

1. Which statement about BGP policies that influence route selection in a multihomed BGP network is true? (Source: Introducing Routing Protocol Tools)
 - A. The default BGP route selection does not always result in optimum routing.
 - B. The default BGP route selection always results in optimum routing.
 - C. After the route selection behavior has been set, it cannot be changed
 - D. The customer receives all routes from both service providers, giving redundancy; therefore, BGP policies are not necessary.

2. Which two potential multihomed network issues can be prevented with IP prefix filters? (Choose two.) (Source: Introducing Routing Protocol Tools)
 - A. the propagation of private AS numbers
 - B. the propagation of private addresses that are used in the network
 - C. the propagation of unreachable next-hop addresses
 - D. the propagation of more specific prefixes from an address range

3. Which three goals represent appropriate reasons to apply AS path filters? (Choose three.) (Source: Introducing Routing Protocol Tools)
 - A. to ensure that only locally originated routes are announced
 - B. to limit routes that are advertised from IBGP neighbors
 - C. to select a subset of all routes, based on their originating AS
 - D. to limit neighbor route updates to specific AS-originated routes
 - E. to ensure that all destination autonomous systems should be received from a specified neighbor
 - F. to change the weight or local preference attributes for all destination autonomous systems

4. Which AS path is matched by the regular expression “72\$”? (Source: Introducing Routing Protocol Tools)
- A. 213 72 218 31 727
 - B. 27 317 271 50 72
 - C. 315 27 723 19 91
 - D. 72 591 368 20 87
5. What is the difference between the regular expressions “_100_” and “_100\$”? (Source: Introducing Routing Protocol Tools)
- A. The first expression refers to routes that have the substring “100” in their AS paths; the second expression refers only to routes that are directly connected to AS 100.
 - B. The first expression refers to routes that have the substring “100” in their AS paths; the second expression refers only to routes that originated in AS 100.
 - C. The first expression refers to routes that go through AS 100; the second expression refers to routes that originated in AS 100.
 - D. The first expression refers to routes that are directly connected to AS 100; the second expression refers to routes that originated in AS 100.
6. How do you implement a **permit all** statement when you are using route maps? (Source: Introducing Route Maps and Routing Policy Language)
- A. By default, a route map has an implicit **permit any** statement if no match is found.
 - B. You must configure a route map with a **permit** parameter and no match clause.
 - C. You must configure a route map with a **deny** parameter and a **deny none** clause.
 - D. You must configure a route map with a **permit any** match clause.
7. What happens to incoming BGP updates that do not match any route map match clauses? (Source: Introducing Route Maps and Routing Policy Language)
- A. They are entered into the BGP table.
 - B. They are entered into the BGP table and marked with a weight of 32768.
 - C. They are not accepted by the router or entered into the BGP table.
 - D. They are entered into the BGP table if a matching route exists in the IP routing table.
8. What are two reasons for using route map sequence numbers? (Choose two.) (Source: Introducing Route Maps and Routing Policy Language)
- A. to allow insertion or deletion of route map entries
 - B. to order the execution sequence of route map match clauses
 - C. to provide an ordered execution sequence for the route map
 - D. to map between prefix list statements and route map match clauses
9. Route maps provide better modularity than RPLs. (Source: Introducing Route Maps and Routing Policy Language)
- A. true
 - B. false

10. Redistribution supports BGP, EIGRP, IS-IS, OSPF, RIP, static, and connected routes. (Source: Implementing Route Redistribution)
- A. true
 - B. false
11. What does the **redistribute eigrp 100 subnets metric-type 1** command do when used in router configuration mode under OSPF process 1? (Source: Implementing Route Redistribution)
- A. redistributes from EIGRP AS 100 into OSPF and changes the metric type from type 2 to type 1
 - B. redistributes from EIGRP AS 100 into OSPF process 1 and changes the metric value of each subnet
 - C. redistributes only from EIGRP AS 100 subnets of metric type 1
 - D. redistributes from EIGRP and processes the first 100 subnets of metric type 1
12. Administrative distance is a way of ranking the trustworthiness of routing information. A higher administrative distance means that a route is more trustworthy. (Source: Implementing Route Redistribution)
- A. true
 - B. false
13. Which routing protocol has the lowest administrative distance? (Source: Implementing Route Redistribution)
- A. OSPF
 - B. external EIGRP
 - C. RIP
 - D. internal EIGRP
14. In which two ways can you avoid routing loops that might result from two-way redistribution? (Choose two.) (Source: Implementing Route Redistribution)
- A. use the **passive-interface** command to suppress routing updates
 - B. use RPLs or route maps to redistribute internal routes only
 - C. use RPLs or route maps to implement route tagging
 - D. use the **no-loop** keyword when redistributing routes
15. What is the difference between local preference and weight? (Source: Influencing BGP Route Selection)
- A. Local preference has a higher priority in BGP path selection.
 - B. Local preference is used AS-wide while weight is local to a single router.
 - C. Local preference is local only to a specific BGP-speaking router.
 - D. Local preference is used to influence incoming path selection.

16. What is the default weight for routes that are received from a BGP neighbor? (Source: Influencing BGP Route Selection)
- A. 0
 - B. 100
 - C. 32768
 - D. depends on the Cisco IOS release
17. Which two statements about the influence of local preference on BGP route selection are true? (Choose two.) (Source: Influencing BGP Route Selection)
- A. When you set local preference, you can view it on neighboring routers, but you must reset it.
 - B. You can use local preference to ensure AS-wide route selection policy.
 - C. Local preference is used to select routes with unequal weights.
 - D. Local preference is the second-strongest criterion in the route selection process.
18. Which option describes AS path prepending? (Source: Influencing BGP Route Selection)
- A. when a router, sending a BGP update, adds the AS number of the router from which it received the route to the AS path attribute
 - B. when a router, sending a BGP update, adds the AS number of the router to which it is sending the route to the AS path attribute
 - C. when a router, sending a BGP update, adds its AS number to the AS path attribute multiple times
 - D. when a router uses the AS path attribute in route selection
19. What are three BGP attributes that are compared before the MED is compared? (Choose three.) (Source: Influencing BGP Route Selection)
- A. largest weight
 - B. originated routes
 - C. AS path length
 - D. lowest IP address
20. What are two reasons why it is not feasible to use the MED to influence return path selection, when multiple autonomous systems are involved? (Choose two.) (Source: Influencing BGP Route Selection)
- A. The MED attribute is designed to influence outbound path selection only.
 - B. The AS path attribute would be used for path selection regardless of any configured MED value.
 - C. The weight attribute will always be used, because it is first in the BGP route-selection process.
 - D. The MED cannot be propagated across several autonomous systems.
21. Does the community attribute have any influence on BGP path selection? (Source: Influencing BGP Route Selection)
- A. No, communities are simply tags that are applied to BGP routes.
 - B. No, communities are nontransitive attributes.
 - C. Yes, BGP paths are selected based on the value in the community tag.
 - D. Yes, the community attribute is part of the BGP route-selection process.

22. What is the result of tagging a route with the no-export community? (Source: Influencing BGP Route Selection)
- A. The route will not be advertised within the local AS.
 - B. The upstream AS will not be allowed to export the route.
 - C. The route cannot be exported to another routing protocol.
 - D. The router will not propagate the route to any external neighbors, except to intraconfederation external neighbors.

Module Self-Check Answers

Answer Key

1. A
2. B, D
3. A, C, D
4. B
5. C
6. B
7. C
8. A, C
9. B
10. A
11. A
12. B
13. D
14. B, C
15. B
16. A
17. B, D
18. C
19. A, B, C
20. B, D
21. A
22. D

Glossary

ABR

area border router. Router located on the border of one or more OSPF areas that connects those areas to the backbone network. ABRs are considered members of both the OSPF backbone and the attached areas. They therefore maintain routing tables describing both the backbone topology and the topology of the other areas.

ACL

access control list. A list kept by routers to control access to or from the router for a number of services (for example, to prevent packets with a certain IP address from leaving a particular interface on the router).

AS

autonomous system. A collection of networks under a common administration sharing a common routing strategy. Autonomous systems are subdivided by areas. An autonomous system must be assigned a unique 16-bit number by the IANA.

ASBR

autonomous system boundary router. An ABR located between an OSPF autonomous system and a non-OSPF network. ASBRs run both OSPF and another routing protocol, such as RIP. ASBRs must reside in a nonstub OSPF area.

ASR

Aggregation Services Router.

BGP

Border Gateway Protocol. Interdomain routing protocol that replaces EGP. BGP exchanges reachability information with other BGP systems. It is defined by RFC 1163.

E2

external type 2

EBGP

Exterior Border Gateway Protocol.

EIGRP

Enhanced Interior Gateway Routing Protocol. Advanced version of IGRP developed by Cisco. Provides superior convergence properties and operating efficiency, and combines the advantages of link-state protocols with those of distance vector protocols.

IBGP

Interior Border Gateway Protocol.

IGP

interior gateway protocol. Internet protocol used to exchange routing information within an autonomous system. Examples of common Internet IGP include IGRP, OSPF, and RIP.

IP

Internet Protocol. Network layer protocol in the TCP/IP stack offering a connectionless internetwork service. IP provides features for addressing, type-of-service specification, fragmentation and reassembly, and security. Defined in RFC 791.

IP address

32-bit address assigned to hosts using TCP/IP. An IP address belongs to one of five classes (A, B, C, D, or E) and is written as 4 octets separated by periods (dotted decimal format). Each address consists of a network number, an optional subnetwork number, and a host number. The network and subnetwork numbers together are used for routing, and the host number is used to address an individual host within the network or subnetwork. A subnet mask is used to extract network and subnetwork information from the IP address. CIDR provides a new way of representing IP addresses and subnet masks. Also called an Internet address.

IPv4

IP version 4. Internet Protocol version 4 is the fourth version in the development of the IP and the first version of the protocol to be widely deployed. Along with IPv6, IPv4 is at the core of standards-based internetworking methods of the Internet. IPv4 is still used to route most traffic across the Internet. IPv4 is a connectionless protocol for use on packet-switched link layer networks (for example, Ethernet). It operates on a best-effort delivery model in that it does not guarantee delivery and does not assure proper sequencing or avoidance of duplicate delivery.

IPv6

IP version 6. Replacement for the current version of IP (version 4). IPv6 includes support for flow ID in the packet header, which can be used to identify flows. Formerly called IPng (next generation).

IS-IS

Intermediate System-to-Intermediate System. OSI link-state hierarchical routing protocol based on DECnet Phase V routing, whereby ISs (routers) exchange routing information based on a single metric to determine network topology.

ISP

Internet service provider. Company that provides Internet access to other companies and individuals.

MED

multi-exit discriminator

NLRI

Network Layer Reachability Information.

OSPF

Open Shortest Path First. Link-state, hierarchical IGP routing algorithm proposed as a successor to RIP in the Internet community. OSPF features include least-cost routing, multipath routing, and load balancing. OSPF was derived from an early version of the IS-IS protocol.

OSPFv3

OSPF version 3

PE

provider edge. Identifies the network devices, under the administrative control of the provider, that connect to CE devices.

RFC

Request for Comments. Document series that is used as the primary means for communicating information about the Internet. Some RFCs are designated by the IAB as Internet standards. Most RFCs document protocol specifications, such as Telnet and FTP, but some RFCs are humorous or historical. RFCs are available online from numerous sources.

RIP

Routing Information Protocol. A distance-vector routing protocol that uses hop count as a routing metric.

RIPng

Routing Information Protocol next generation

RIPv2

Routing Information Protocol Version 2

RPL

routing policy language

TCP/IP

Transmission Control Protocol/Internet Protocol. Common name for the suite of protocols developed by the U.S. DoD in the 1970s to support the construction of worldwide internetworks. TCP and IP are the two best-known protocols in the suite.

TTL

Time to Live. A mechanism that limits the lifespan or lifetime of data in a computer or network.

UNIX

Operating system developed in 1969 at Bell Laboratories. UNIX has gone through several iterations since its inception. These include UNIX 4.3 BSD (Berkeley Standard Distribution), developed at the University of California at Berkeley, and UNIX System V, Release 4.0, developed by AT&T.

WAN

wide-area network. Data communications network that serves users across a broad geographic area and often uses transmission devices provided by common carriers. Frame Relay, SMDS, and X.25 are examples of WANs.

