

BGP

Configuring BGP on Cisco Routers

Version 4.0

Student Guide



Credits

Copyright notices:



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 U.S. 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.		

Copyright Date:

© 2015 Cisco Systems, Inc.

Welcome Students



Students, this letter describes important course evaluation access information.

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

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

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

Sincerely,
Cisco Systems Learning

Course Introduction

Overview

In this course you will learn how to design, implement, and verify a BGP in the customer and service provider networks.

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

- Configure, monitor, and troubleshoot basic BGP to enable interdomain routing in a network scenario with multiple domains
- Use BGP policy controls to influence the route selection process with minimal impact on BGP route processing in a network scenario where you must support connections to multiple ISPs
- Use BGP attributes to influence the route selection process in a network scenario where you must support multiple connections
- Complete the correct BGP configuration to successfully connect the customer network to the Internet in a network scenario in which multiple connections must be implemented
- Configure the service provider network to behave as a transit AS in a typical implementation with multiple BGP connections to other autonomous systems
- Enable route reflection as possible solution to BGP scaling issues in a typical service provider network with multiple BGP connections to other autonomous systems
- Use available BGP tools and features to optimize the scalability of the BGP routing protocol in a typical BGP network

Learner Skills and Knowledge

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

- Intermediate to advanced knowledge of Cisco IOS Software configuration
- Configuring and troubleshooting RIP, EIGRP, OSPF and IS-IS
- Skills and knowledge equivalent to those learned in:
 - *Interconnecting Cisco Networking Devices v2.0, Part 1* (ICND1 v2.0) and *Part 2* (ICND2 v2.0), or
 - *Interconnecting Cisco Networking Devices: Accelerated Version 2.0* (CCNAX v2.0)
 - *Implementing Cisco IP Routing* (ROUTE v2.0)
 - *Building Cisco Service Provider Next-Generation Networks Part 1* (SPNGN1) v1.2
 - *Building Cisco Service Provider Next-Generation Networks Part 2* (SPNGN2) v1.2

Course Goal and Objectives

This topic describes the course goal and objectives.

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

- Describe how to configure, monitor, and troubleshoot basic BGP to enable interdomain routing in a network scenario with multiple domains
- Describe how to use BGP policy controls to influence the BGP route selection process in a network scenario in which you must support connections to multiple ISPs
- Describe how to use BGP attributes to influence the route selection process in a network scenario where you must support multiple connections.
- Describe how to successfully connect the customer network to the Internet in a network scenario in which multiple connections must be implemented

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

- Describe how to configure the service provider network to behave as a transit AS in a typical implementation with multiple BGP connections to other autonomous systems.
- Enable route reflection as possible solution to BGP scaling issues in a typical service provider network with multiple BGP connections to other autonomous systems
- Describe the available BGP tools and features to optimize the scalability of the BGP routing protocol in a typical BGP network

The goal of the course is to provide learners with in-depth knowledge of BGP.

Course Flow

This topic presents the suggested flow of the course materials.

	AM	PM
Day 1	Course Intro Module 1: BGP Overview	Module 1 (Cont.)
Day 2	Module 2: BGP Transit Autonomous Systems	Module 3: Route Selection Using Policy Controls
Day 3	Module 3 (Cont.)	Module 4: Route Selection Using Attributes
Day 4	Module 4 (Cont.) Module 5: Customer-to-Provider Connectivity with BGP	Module 5 (Cont.)
Day 5	Module 6: Scaling Service Provider Networks Module 7: Optimizing BGP Scalability	Module 7 (Cont.)

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

Your Training Curriculum

This topic presents the training curriculum for this course.



You are encouraged to join the Cisco Certification Community, a discussion forum open to anyone holding a valid Cisco Career Certification (such as Cisco CCIE®, CCNA®, CCDA®, CCNP®, CCDP®, CCIP®, CCNP® Security and CCNP® Voice, or CCSP™). It provides a gathering place for Cisco certified professionals to share questions, suggestions, and information about Cisco Career Certification programs and other certification-related topics. For more information, visit <http://www.cisco.com/web/learning/training-index.html>.

Training Curriculum

This subtopic presents the learning center information.



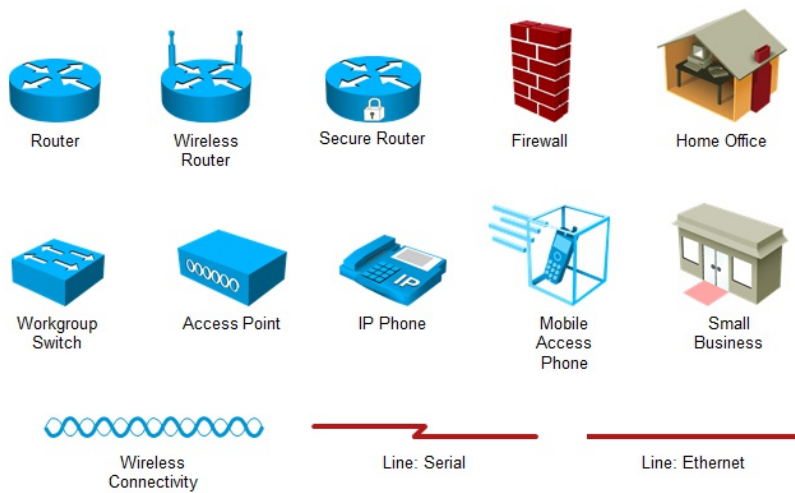
Additional information is available at:

- https://learningnetwork.cisco.com/community/learning_center
- <https://learningnetwork.cisco.com/community/certifications/ccna>
- Your name
- Your company
- Job responsibilities
- Skills and knowledge
- Brief history
- Objective



Additional References

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



Cisco Glossary of Terms

For additional information on Cisco terminology, refer to the Cisco Internetworking Terms and Acronyms glossary of terms at [http://docwiki.cisco.com/wiki/Category:Internetworking_Terms_and_Acronyms_\(ITA\)](http://docwiki.cisco.com/wiki/Category:Internetworking_Terms_and_Acronyms_(ITA)).

BGP Overview

Introduction

BGP is an interdomain (interautonomous system) routing protocol that is used to exchange routing information for the Internet. BGP, by design, is a very robust and scalable routing protocol. Because BGP is deployed as an interdomain routing protocol, it has many rich features that allow you to implement a variety of routing policies.

In this module you will learn the basic BGP technology, details about BGP session establishment and routing information exchange. You will also learn the basic Cisco IOS BGP configuration and troubleshooting tasks.

Upon completing this module, you will be able to:

- Defines how to identify appropriate BGP use and limitations.
- Describe BGP path attributes and the functionality of each attribute.
- Describe BGP neighbors and neighbor session establishment procedures.
- Describe BGP route processing and BGP routing updates
- Describe how to configure basic BGP
- Describe how to perform the steps to correct basic BGP configuration and session errors

Introducing BGP

Overview

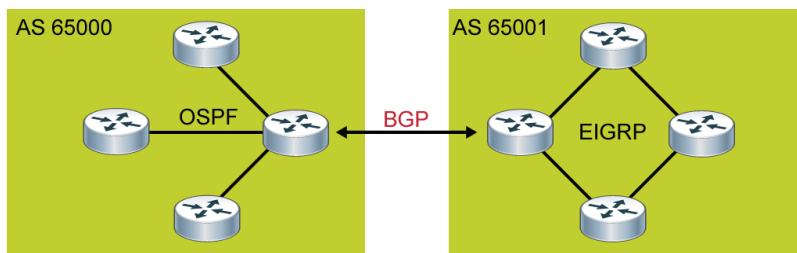
The [BGP](#) is a very robust and scalable routing protocol. You can see that from the fact that it is the routing protocol that is used on the Internet. Service providers and customer networks, such as universities and corporations, usually use an [IGP](#) such as [EIGRP](#), [OSPF](#), or [IS-IS](#) for the exchange of routing information within their networks. Any communication between these IGPs and the Internet or between service providers will be accomplished through BGP. This lesson introduces basic BGP characteristics and features.

Upon completing this lesson, you will be able to:

- Describe interdomain routing in relation to the design goals of interdomain routing protocols
- Explain the need for an interdomain routing protocol
- List the basic characteristics of BGP
- List BGP development considerations
- Identify when a single-homed customer should use BGP as an interdomain routing protocol
- Describe when BGP is appropriate for the multihomed customer
- Describe the use of BGP in a transit autonomous backbone
- List some of the limitations of BGP

Interdomain Routing

When talking to people who are involved with Internet routing, network administrators commonly use the terms "autonomous system," "interdomain routing," "interior routing protocol," and "exterior routing protocol."



- An AS is a collection of networks under a single technical administration.
- An IGP is run inside an AS, resulting in optimum intra-AS routing.
- An EGP is run between autonomous systems to enable routing policies and improve security.

These terms, which may be confusing for a novice, are defined as follows:

- An [AS](#) is a collection of networks under a single technical administration. Other definitions refer to a collection of routers or [IP](#) prefixes, but in the end they are all essentially the same thing. The important principle is the technical administration, which means sharing the same routing protocol and routing policy. Legal and administrative ownership of the routers does not matter with autonomous systems. Autonomous systems are identified by AS numbers. AS numbers are 16-bit, unsigned integers ranging from 1 to 65,535. Public AS numbers (1 to 64,511) are assigned and managed by an Internet registry. A range of private AS numbers (64,512 to 65,535) has been reserved for customers that need an AS number to run [BGP](#) in their private networks. New 32-bit AS numbers were created when the AS number pool from [IANA](#) approached exhaustion.
- Interdomain routing is routing between autonomous systems. It is usually based on a set of policies, not just the technical characteristics of the underlying infrastructure.
- Exterior routing protocols (BGP being the only exterior routing protocol that is used today) are protocols that have the right set of functions to support various interdomain routing policies. Such protocols are contrary to interior routing protocols (for example, [OSPF](#), [IS-IS](#), or [EIGRP](#)). Interior routing protocols focus only on finding the optimum (usually fastest) route between two points, without respect to routing policies.

Design Goals for Interdomain Routing

Scalability

- The Internet has more than 600,000 aggregated routes and is still growing.

Secure routing information exchange

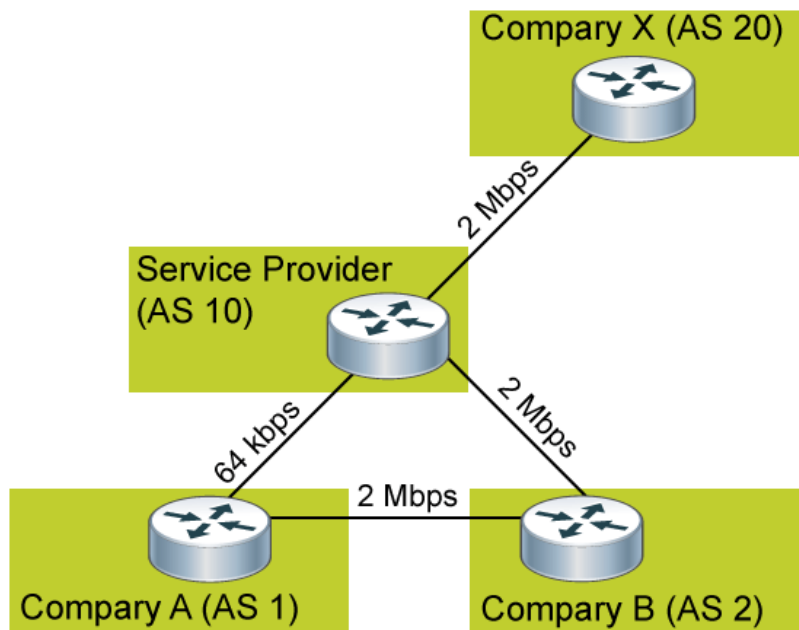
- Routers from another AS cannot be trusted.
- Tight filters are required; authentication is desirable.

Support for routing policies

- Routing between autonomous systems might not always follow the optimum path.
- Exterior routing protocols have to support a wide range of customer requirements.

Why External Routing Protocols?

The example illustrates the need for an interdomain routing protocol. It depicts two companies that are connected to the Internet via leased lines of differing speeds.



- Assuming standard IGP route selection rules, how will the traffic between AS 1 and AS 20 flow?
- Will AS 2 allow this traffic?
- How would you solve this problem with OSPF or EIGRP?

In routing protocols other than BGP, routing decisions are normally made to take advantage of the highest bandwidth available. Doing so would make traffic between AS 1 and AS 20 flow via AS 2. This situation would allow the users in Company A to generate traffic on the Internet access line that was purchased and paid for by Company B. So, this situation is not desirable for AS 2.

Company B is unlikely to allow traffic from Company A to reach the Internet using the Company B access line. Company B, in fact, could create an access-list blocking all IP packets from AS 1 from being transmitted on the 2-Mbps serial line from Company B to the Internet. That action would create a black hole because Company A would send its packets to Company B and then Company B would drop them.

To avoid this situation, Company B must make sure that the packets from Company A that are destined for the Internet are never sent to Company B. Also, Company B must make sure that packets from the Internet that are destined for Company A are never sent using the Internet access line to Company B. Company B could implement a routing policy that indicates that AS 2 will receive reachability information from AS 1 for its own use but that AS 2 will not forward that particular information to the Internet. Also, AS 2 will receive reachability information about the Internet from its [ISP](#) but will never forward that information to AS 1. Only networks local to AS 2 will be sent to AS 1.

The result of this routing policy would be that AS 1 sees all networks within AS 2 as reachable over the 2-Mbps link that directly connects AS 1 with AS 2. The routers in AS 1 will not see the rest of the Internet as reachable through AS 2. Therefore, AS 1 forwards packets toward the Internet directly over the 64-kbps link.

Also, the IP networks in AS 1 will appear reachable by AS 2 over the 2-Mbps link, which directly connects AS 1 with AS 2. However, the ISP will not receive that reachability information from AS 2; it will receive it only from AS 1. Therefore, traffic from the Internet to Company A will be transmitted over the 64-kbps link.

This routing policy is easy to implement when network administrators are using BGP but impossible to implement with any other routing protocol. EIGRP, for example, can do route filtering only on individual IP subnets, not on all prefixes belonging to an AS. Link-state protocols, such as OSPF, cannot do powerful route filtering at all. BGP uses AS numbers to do route filtering. This approach makes it possible to scale BGP over Internet.

BGP Characteristics

BGP is a distance vector protocol. So, BGP announces to its neighbors those IP networks that it can reach itself. The receivers of that information say, "If that AS can reach those networks, then I can reach them via the AS."

- BGP is a distance vector protocol with enhancements:
 - Reliable updates
 - Triggered updates only
 - Rich metrics (called path attributes)
- BGP is designed to scale to huge internetworks

If two different paths are available to reach the same IP subnet, then the shortest path is used. This determination requires a mechanism capable of measuring the distance. All distance vector protocols have such mechanisms, called "metrics." BGP contains a very sophisticated method of computing the shortest path by using attributes that are attached to the reachable IP subnet.

BGP sends routing updates to its neighbors by using a reliable transport. This technique means that the sender of the information always knows that the receiver has actually received the information. As a result, there is no need for periodic updates or routing information refreshes. In BGP, only information that has changed is transmitted.

BGP enables reliable information exchange and is capable of batching the routing updates. These two characteristics allow BGP to scale to large, Internet-sized networks.

Reliable updates:

- TCP used as transport protocol
- No periodic updates
- Periodic keepalives to verify TCP connectivity
- Triggered updates are batched and rate-limited
 - Every 5 seconds for internal peer
 - Every 30 seconds for external peer

The reliable transport mechanism that BGP uses is standard TCP. BGP is an application protocol that uses both the TCP and IP protocols for reliable connections.

Because BGP uses a reliable transport, the sender knows that the receiver has actually received the transmitted information. This capability makes periodic updates unnecessary.

A router that has received reachability information from a BGP peer must be sure that the peer router is still there. Otherwise, the router could route traffic toward a next-hop router that is no longer available, causing the IP packets to be lost in a black hole. TCP does not provide the service to signal that the TCP peer has been lost, unless some application data is actually transmitted between the peers. In an idle state, where there is no need for BGP to update its peer, the peer could be unreachable without TCP detecting it. Therefore, BGP takes care of detecting the presence of neighbors by periodically sending small BGP keepalive packets to them. These packets are considered application data by TCP and therefore must be transmitted reliably. According to the BGP specification, the peer router also must reply with a BGP keepalive packet.

When BGP was created, a key design goal was to be able to handle enormous amounts of routing information in very large and complex networks. In this environment, many links could go up and down (flapping), causing topology changes. The routing protocol must consider all these changes. But low convergence time and quick responses to topology changes require fast updates and high CPU power to process both incoming and outgoing updates. The larger the network, the more updates per second can be expected if immediate response is required. The presence of too many updates in large networks can jeopardize network scalability.

The designers of BGP decided that scalability was a more important issue than low convergence time, so BGP was designed to batch updates. Any changes that are received within the batch interval time are saved. At the end of the interval, only the remaining result is forwarded in an outgoing update. If a network flaps several times during the batch interval, only the state at the end of the interval is sent in an update.

The batching feature avoids an uncontrolled flood of updates all over the Internet because the batching procedure limits the number of updates.

Common BGP uses:

- Customers that are connected to more than one service provider
- Service provider networks (transit autonomous systems)
- Service providers exchanging traffic at an exchange point (CIX, GIX, NAP, ...).
 - NAP defines the exchange points between region and core.
 - CIX or GIX points define international exchange points.
- Network cores of large-enterprise customers

NAP—Network Access Point, CIX—Commercial Internet eXchange, GIX—Global Internet eXchange

BGP Development Considerations

The designers of the BGP protocol have succeeded in creating a highly scalable routing protocol, which can forward reachability information between autonomous systems (also known as routing domains). The designers had to consider an environment with an enormous number of reachable networks and complex routing policies that were driven by commercial rather than technical considerations.

- BGP was designed to perform well in the following areas:
 - Interdomain routing applications
 - Huge internetworks with large routing tables
 - Environments that require complex routing policies
- Some design tradeoffs were made:
 - BGP uses TCP for reliable transport—CPU-intensive
 - Scalability is the top priority—slower convergence

TCP, a well-known and widely proven protocol, was chosen as the transport mechanism. That decision kept BGP simple, but it increased the CPU resource requirements for routers running BGP. The point-to-point nature of TCP also introduces a slight increase in network traffic. This increase occurs because any update that should be sent to many receivers has to be multiplied into several copies. Those copies are then transmitted on individual TCP sessions to the receivers.

Whenever there was a design choice between fast convergence and scalability, scalability was the top priority. The batching of updates and the relatively low frequency of keepalive packets are examples of designers placing convergence time second to scalability.

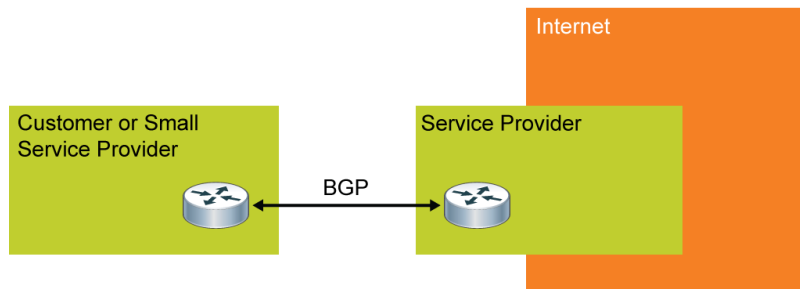


BGP convergence times can be modified with the configuration of nondefault values for BGP scan and advertisement timers.

Single-Homed Customers

Normal Internet access to a single [ISP](#) does not require [BGP](#); static routes are more commonly used to handle this situation. Small ISPs that buy Internet connectivity from other ISPs use this type of connectivity more often. Especially if they want to start their business the proper way—by using their own [AS](#) number and having their own address space.

Large customer or small ISP connecting to the Internet



The figure shows a customer network that is connected to the Internet using a single ISP, but such a scenario is generally not the case when BGP is used.

Use Guidelines—Single-Homed Customers

Use BGP between the customer and the service provider in these situations:

- Customers multihomed to the same service provider
- Customers that need dynamic routing protocol with the service provider to detect failures
 - Hint: Use private AS number for these customers.
- Smaller ISPs that need to originate their routes in the Internet

Use static routes in all other cases:

- Static routes always simpler than BGP

Under certain conditions, BGP must be configured between the customer and the service provider. For example, BGP is needed when customers are multihomed to the same service provider, that is, the customer networks have multiple links connecting them with the service provider network. Such customers require dynamic routing protocol interaction with the service provider to detect link failures. Private AS numbers (AS numbers between 64,512 and 65,534 or between 4,200,000,000—4,294,967,294) are usually implemented in BGP configurations for these customers.

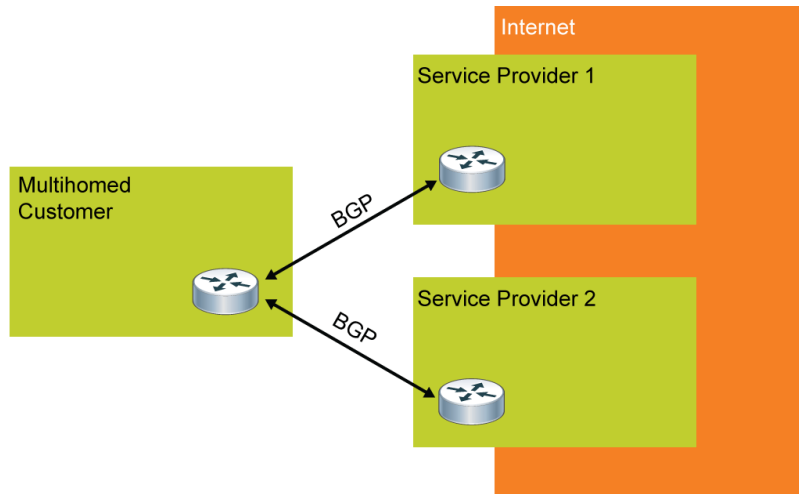
Customers that plan to connect to more than one ISP, and small ISPs that plan to have multiple Internet connections in the future, usually use BGP with their service provider. They use this option even when they have a single link with the service provider in order to be prepared for future upgrades.

In all other cases, using static routes from the service provider toward the customer and using a default static route from the customer toward the service provider is the preferred method of provider-to-customer routing in the Internet.

Multihomed Customers

The **BGP** is appropriate for the multihomed customer. The customer must have its own officially assigned **AS** number. The customer is also responsible for announcing its own **IP** networks to both **ISPs**. Both ISPs forward all routes that they receive from the Internet to the customer network. The customer should avoid forwarding any routing information that they receive from one ISP to the other. Otherwise, the customer becomes a transit provider between the two ISPs. Most customers like to avoid this situation because it creates a resource drain on routers and network links.

Customer connecting to more than one service provider



The figure illustrates a customer network that is connected to two different ISPs, requiring the use of BGP for full redundancy.

Full redundancy is achieved in this setup. If either of the two access links fails, the reachability information that was previously transmitted on the now-failed link is withdrawn. But BGP reachability information is still announced by the customer router over the remaining link. Thus, the ISP still sees all networks within the customer AS as reachable but only over the remaining path. Also, received routes from the Internet are withdrawn when the link fails but routes that are received over the remaining link are not affected. Thus, the Internet, including the ISP to which the direct connection has failed, is still reachable over the remaining link.

This design can also handle other problems. A case where both access links are available, but the connection between one of the ISPs and the rest of the Internet is lost, works as follows. The ISP that has a problem reaching the rest of the Internet withdraws all those routes and tells the customer AS that it can no longer reach the Internet. But the networks local to the ISP with the Internet reachability problem are still reachable by the customer, so those routes are not withdrawn. The networks in the customer AS are still reachable by the ISP in trouble, but that ISP can no longer forward the announcement to the rest of the Internet. The rest of the Internet will, however, see the customer networks as reachable over the path to the other ISP, which is fully functional.

User Guidelines—Multihomed Customers

- BGP is almost mandatory for multihomed customers.
- Multihomed customers have to use public AS numbers.
- Multihomed customers should use a provider-independent address space.

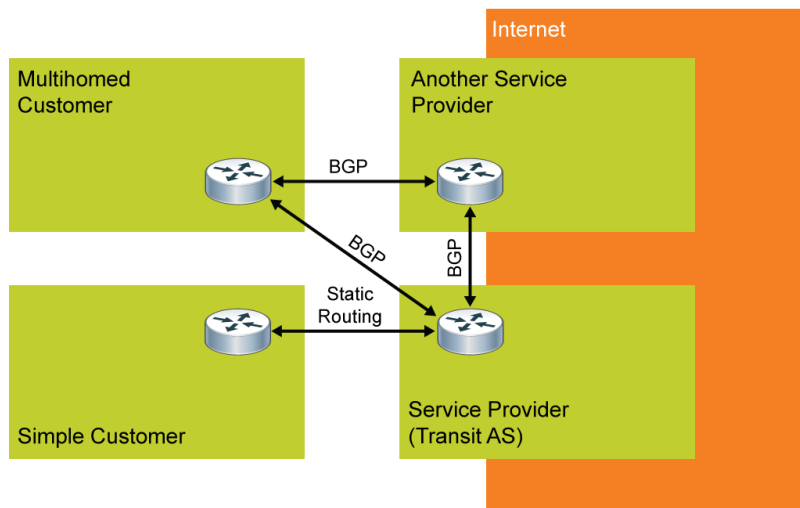
The following use guidelines apply to multihomed customers:

- Although there are designs where BGP could be avoided, most multihomed customers need to use BGP with their service providers.
- Multihomed customers must have their own AS numbers. It is recommended to use a public AS number (between 1 and 23.455, between 23.457 and 64.534, or between 131.072 and 4.199.999.999).
- Multihomed customers should use a provider-independent address space, which is allocated to them directly by an Internet registry.

Transit Autonomous Systems

[BGP](#) is most commonly implemented in service provider networks to ensure connectivity between customers and the rest of the Internet. An [ISP](#) might exchange BGP updates with the customers or use static routing toward them. That ISP also connects to other ISPs and is required to forward the routes that are received from customers to the rest of the Internet, as well as in the other direction. As a result, user data traffic starts to flow between the customers and the rest of the Internet. Such a network, providing transit services to traffic that is originated in other networks, is called a "transit autonomous system," or "transit [AS](#)." A transit AS is an AS that exchanges BGP routing information with other autonomous systems and forwards information that is received from one AS to another AS.

Using BGP to exchange routes is mandatory for transit autonomous systems (provider networks carrying customer traffic).



When routing information is forwarded, the receiver will see an available path to a destination and start transmitting user data toward the destination using that path. The transit AS must be prepared to relay the user data.

ISP networks can sometimes have dedicated peer-to-peer connections. These connections are sometimes called private peering. ISPs also interconnect at exchange points. Technically, an exchange point is just a multiaccess subnet: a [LAN](#) (for example, a [Gigabit Ethernet](#) or [Fast Ethernet](#) switch), a [DPT](#) ring, or an [ATM](#) switch. Many ISPs can connect to an exchange point and establish BGP sessions.

The benefit of an exchange point is that it is highly scalable. There is no need for more physical interfaces in the ISP border router when a new ISP is launched. If the already established ISPs want to, they can open a BGP session with the new ISP. When this session is opened, they start to exchange routing information and then user data traffic over the exchange point.

BGP Limitations

BGP-enabled routers make forwarding decisions based on the destination [IP address](#) only. The source IP address does not affect the BGP decision.

BGP and associated tools cannot express all routing policies.

- You cannot influence the routing policies of downstream autonomous systems.

“BGP does not enable one AS to send traffic to a neighbor AS intending that the traffic take a different route from that taken by traffic originating in the neighbor AS.” (RFC 1771)

If an [AS](#) acts as a transit AS for other autonomous systems, the [IP](#) packets that are created and transmitted from the other autonomous systems are not treated differently from the IP packets that are created and transmitted from the local AS. If the local AS has decided that the best path to reach a certain destination is via a specific next-hop router, then it will route all user data traffic toward the final destination via that specific next-hop router. The local AS makes its decision based on destination address only, regardless of which IP host has sourced the IP packets.

Summary

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

- BGP is an enhanced distance vector protocol with reliable transport.
- Customers that plan to connect to more than one ISP, and small ISPs that plan to have multiple Internet connections in the future, usually use BGP with their service provider. Most multihomed customers use BGP with their service providers.
- A transit AS is an AS that exchanges BGP routing information with other autonomous systems and forwards information that is received from one AS to another AS.
- In the BGP, you cannot influence the routing policies of downstream autonomous systems.

Understanding BGP Path Attributes

Overview

To aid routers in calculating the best route to select when multiple paths to a particular destination exist, routes that are learned via [BGP](#) have properties that are associated with them. These properties are referred to as BGP path attributes. An understanding of how BGP path attributes influence route selection is required to design robust BGP networks.

Upon completing this lesson, you will be able to:

- Describe the concept of BGP path attributes
- Explain the difference between mandatory and discretionary well-known BGP attributes
- Explain the difference between nontransitive and transitive optional BGP attributes
- Describe the functionality of the AS-path attribute
- Describe the functionality of the next-hop attribute

BGP Path Attributes

Each BGP update consists of one or more IP subnets and a set of attributes that are attached to them.

- BGP metrics are called path attributes.
- BGP attributes are categorized as "well-known" and "optional."
- All compliant implementations must recognize well-known attributes.
- You should not expect that all implementations can recognize optional attributes. Only some implementations (could be private) are able to recognize optional attributes.

All BGP implementations are required to recognize certain attributes. Those attributes are called "well-known BGP attributes."

Attributes that are not well-known are called "optional." Optional attributes are either specified in a later extension of BGP or even in private vendor extensions that are not documented in a standard document.

Well-Known BGP Attributes

Well-known attributes are divided into mandatory and discretionary.

- All well-known attributes are propagated to other neighbors.
- The difference between mandatory and discretionary well-known attributes:
 - **Mandatory** well-known attributes must be present in all update messages.
 - **Discretionary** well-known attributes are optional; they could be present in update messages.

There is a small set of three specific well-known attributes that are required to be present on every update. These attributes are the next-hop, [AS-path](#), and origin attributes and are referred to as "mandatory well-known attributes."

Other well-known attributes may or may not be present, depending on the circumstances under which the updates are sent and the desired routing policy. The well-known attributes that could be present, but are not required, are called "discretionary well-known attributes."

When a router receives a [BGP](#) update, it analyzes the attached attributes and compares them with the attributes that were attached to the same [IP](#) subnet when it was received from a different source. The router then makes a decision about which source indicates the best path to the particular IP subnet. The best route is propagated, along with its well-known attributes, to other BGP-speaking neighbors.

Mandatory Well-Known BGP Attributes

- Origin
 - The origin of a BGP route

i	Route originated in an IGP
e	Route originated in EGP
?	Route was redistributed into BGP

- AS-path
 - Sequence of AS numbers through which the network is accessible
- Next-hop
 - IP address of the next-hop router

The three mandatory well-known attributes are origin, AS-path, and next-hop:

- **Origin:** When a router first originates a route in BGP, it sets the origin attribute. If information about an IP subnet is injected using the **network** command or via aggregation (route summarization within BGP), the origin attribute is set to "i" for [IGP](#). If information about an IP subnet is injected using redistribution, the origin attribute is set to "?" for unknown or incomplete information (these two words have the same meaning). The origin code "e" was used when the Internet was migrating from [EGP](#) to BGP and is now obsolete.
- **AS-path:** The egress router modifies the AS-path attribute every time information about a particular IP subnet passes over an AS border. When a router first originates a route in BGP, the AS-path attribute is empty. Each time that the route crosses an AS boundary, the transmitting AS prepends its own AS number to appear first in the AS path. You can track the sequence of autonomous systems through which the route has passed by using the AS-path attribute.
- **Next-hop:** The router also modifies the next-hop attribute as the route passes through the network. This attribute indicates the [IP address](#) of the next-hop router. The next-hop router is the router to which the receiving router should forward the IP packets to reach the destination that is advertised in the routing update.

Discretionary Well-Known BGP Attributes

- Local preference
 - Used for consistent routing policy within AS
- Atomic aggregate
 - Informs the neighbor AS that the originating router aggregated routes

All BGP implementations must support discretionary well-known attributes. However,

discretionary well-known attributes do not have to be present in all BGP updates. Routers use discretionary well-known attributes only when those functions are required.

The following are descriptions of these two attributes:

- **Local preference:** Local preference is used in the route selection process. This attribute is carried within an AS only. The router prefers a route with a high local preference value to a route with a low value. By default, routes that are received from a peer AS are tagged with the local preference set to a value of 100 before they are entered into the local AS. If this value is changed through BGP configuration, the BGP selection process is influenced. Because all routers within the AS get the attribute along with the route, a consistent routing decision is made throughout the AS.
- **Atomic aggregate:** The atomic aggregate attribute is attached to a route that is created as a result of route summarization (called "aggregation" in BGP). This attribute signals that information that was present in the original routing updates may have been lost when the updates were summarized into a single entry.

Optional BGP Attributes

Optional [BGP](#) attributes are transitive or nontransitive.

- The difference between transitive or nontransitive optional BGP attributes:
 - **Transitive optional attributes:** Propagated to other neighbors if not recognized; partial bit set to indicate that the attribute was not recognized
 - **Nontransitive optional attributes:** Discarded if not recognized
- Recognized optional attributes are propagated to other neighbors based on their meaning (not constrained by transitive bit).

When a router receives an update that contains an optional attribute, the router checks to see whether its implementation recognizes the particular attribute. If it does, then the router should know how to handle it and whether to propagate it.

If the router does not recognize the attribute, the BGP implementation should look for the transitive bit in the attribute code. Some attributes, although not recognized by the router, might still be helpful to upstream routers and should be propagated. These attributes (called "transitive optional attributes") are propagated even when they are not recognized. If a router propagates an unknown transitive optional attribute, it sets an extra bit in the attribute header. This bit is called the "partial bit." The "partial bit" indicates that at least one of the routers in the path did not recognize the meaning of a transitive optional attribute.

Other attributes, called "nontransitive optional attributes," might be of no value to upstream routers if a router earlier in the path does not recognize them. Routers that do not recognize these attributes drop them.

Nontransitive attributes

- MED:
 - Used to discriminate between multiple entry points to a single AS

Transitive attributes

- Aggregator
 - Specifies IP address and AS number of the router that performed route aggregation
- Community
 - Used for route tagging

One of the nontransitive optional attributes is the [MED](#) attribute, which also influences the BGP route selection process. Whenever there are several links between two adjacent autonomous systems, one [AS](#) can use the MED attribute to tell another AS to prefer one of the links for specific destinations.

Transitive optional attributes include the following:

- **Aggregator:** Identifies the AS and the router within that AS that created a route summarization, or aggregate.
- **Community:** A numerical value that can be attached to certain routes as they pass a specific point in the network. For filtering or route selection purposes, other routers can examine the community value at different points in the network. BGP configuration may cause routes with a specific community value to be treated differently than others.

AS-Path Attribute

An edge router modifies the AS-path attribute every time information about a particular IP subnet passes over an AS border. When a router first originates a route in BGP, the AS-path attribute is empty. The local AS number is prepended to the AS path each time that the route crosses an AS boundary.

- The AS-path attribute is empty when a local route is inserted in the BGP table.
- The AS number of the sender is prepended to the AS-path attribute when the routing update crosses AS boundary.
- The receiver of BGP routing information can use the AS-path attribute to determine through which AS the information has passed.
- An AS that receives routing information with its own AS number in the AS path silently ignores the information.

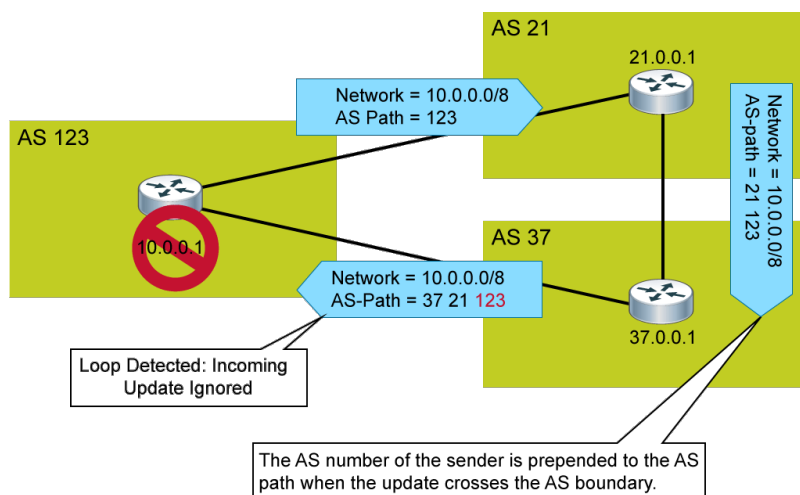
There are several consequences of AS-path attribute behavior:

- When you examine BGP routes, the AS path can be interpreted as the sequence of autonomous systems that must be passed through to reach the indicated network. The AS that originally injected the route into BGP is always found at the rightmost end of the AS path.
- It is easy to distinguish local routes from routes that have been received from other autonomous systems. BGP routes with an empty AS path were injected into BGP from within the local AS.

The AS-path attribute is also used to avoid routing loops. When a router receives a BGP update, it checks the AS-path attribute and looks for its own AS number. If that number is found in the AS path, then the route has already crossed the local AS and the router is now faced with a routing information loop. To avoid this situation, the route is silently ignored.

AS-Path Attribute Example

The figure shows how BGP loop prevention works.



The network 10.0.0.0/8 is local to AS 123. The router in AS 123 injects the route 10.0.0.0/8 into BGP with an empty AS-path attribute.

When the edge router in AS 123 sends a routing update about network 10.0.0.0/8 to AS 21, the AS number 123 is prepended to the empty AS path, resulting in an AS path consisting of only 123. The sending router does the prepending as part of the outgoing BGP update processing. While the route is still within AS 123, the AS-path entry for AS 123 does not appear in the AS path.

The router in AS 21 propagates the information about the network 10.0.0.0/8 to AS 37. Because it is sending the BGP update to AS 37, it prepends its own AS number to the AS path, resulting in an AS path consisting of the sequence of 21 123.

AS 37 also propagates the received route to AS 123. To avoid a routing loop, where AS 123 might try to reach its own network (10.0.0.0/8) via AS 37, BGP has a built-in mechanism that causes the router in AS 123 to drop the incoming update as soon as it finds its own AS 123 in the AS path. No error will be signaled, because nothing is really wrong. It is merely the procedure that BGP uses to avoid a routing information

loop.

Next-Hop Attribute

The [BGP](#) next-hop attribute identifies the [IP address](#) that a router should use to forward packets toward the destination that is announced in a BGP routing update.

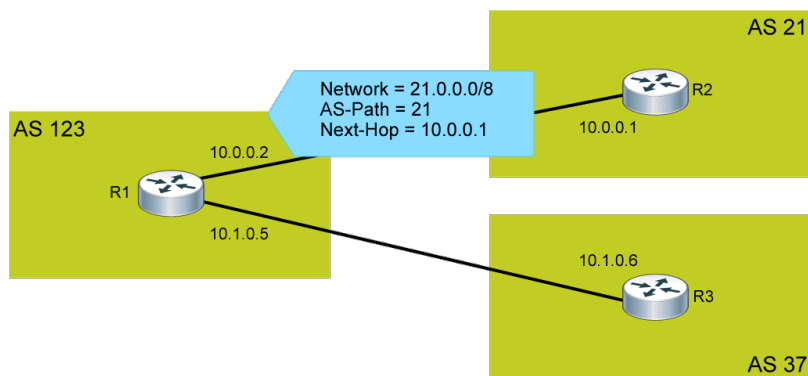
- The next-hop attribute indicates the next-hop IP address that is used for packet forwarding.
- It is usually set to the IP address of the sending EBGp router.
- Can be set to a third-party IP address to optimize routing.

In most cases, the sending router sets the next-hop attribute to its own IP address. There are cases, however, where the next-hop IP address points to a third router.

Next-Hop Processing

The figure shows the usual next-hop processing.

- Next-hop attribute is usually set to the IP address of the sending router.

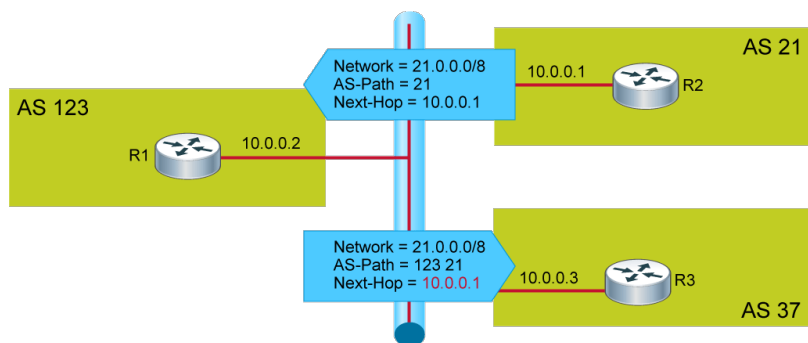


The processing is as follows:

1. R2 announces network 21.0.0.0/8 to R1. The outgoing IP address of R2 (the address that is used to establish the BGP [TCP](#) session) is used as the BGP next hop.
2. R1 receives the routing update and installs it in its BGP table and routing table. Should R1 need to forward packets toward network 21.0.0.0/8, it would send those packets toward the IP address 10.0.0.1 (R2).
3. When R1 propagates the information about 21.0.0.0/8 to R3, it sets the BGP next-hop attribute to its own IP address.

Next-Hop Processing on Shared Media

- If the receiving BGP router is in the same subnet as the current next-hop address, the next-hop address remains unchanged to optimize packet forwarding.



The next-hop processing changes if the BGP routers connect to a shared subnet. In the figure here, if R1 announces the network 21.0.0.0/8 to R3 with the BGP next-hop address set to R1, the packets from AS 37 toward network 21.0.0.0/8 will have to cross the shared [LAN](#) twice. R1 thus sends the routing update toward R3 with the BGP next-hop address unchanged (still pointing toward R2), allowing optimal data transfer across the shared LAN.

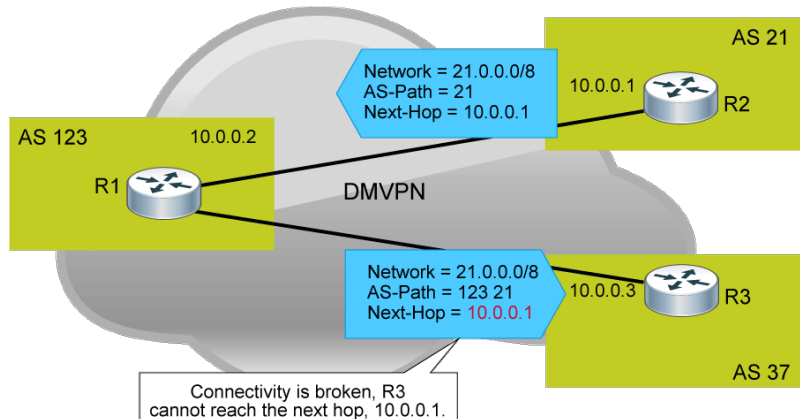
More formally, the BGP next-hop rule states that if the



current BGP next hop is in the same IP subnet as the receiving router, the next-hop address is not changed; otherwise, the next-hop attribute is changed to the IP address of the sending router.

Next-Hop Processing on NBMA Network

- BGP next-hop processing can break connectivity with improper network designs over partially meshed WAN networks.



BGP next-hop processing results in optimum data transfer over shared media (for example, a LAN subnet). In partially meshed networks (such as [DMVPN](#)), BGP next-hop processing can break IP connectivity.

Consider, for example, the network diagram in the figure. R1 sends a routing update about network 21.0.0.0/8 to R3 with R2 set to the next-hop address (as they are all in the same subnet). Because there is no direct connection (virtual circuit) between R3 and R2 but R3 still tries to send packets directly toward R2, the connectivity between AS 37 and AS 21 is broken.

There are two ways to solve this connectivity loss:

- Use the subinterfaces on R1 to make sure that R2 and R3 are in different subnets. BGP next-hop processing will then ensure that R1 is the BGP next-hop in the outgoing BGP updates.
- Disable the BGP next-hop processing on R1 in an existing multipoint DMVPN design that shares a common subnet. (This option is strongly discouraged in normal BGP design because routing problems should be solved with a proper network design of point-to-point subinterfaces.)

Summary

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

- BGP metrics that are attached to a BGP route are called "path attributes." Some path attributes are well-known. Every BGP implementation should be able to recognize the well-known attributes.
- Some of the well-known attributes are mandatory and have to be present in every BGP update. Mandatory well-known attributes are the AS-path, next-hop, and origin. Other well-known attributes are discretionary.
- Attributes that are not required to be recognized by every BGP implementation are called "optional." These attributes could be transitive (propagated if not recognized) or nontransitive (dropped).
- The AS-path attribute lists the autonomous systems that the routing update has already crossed. This attribute is used for BGP loop detection and BGP route selection.
- The next-hop attribute specifies the IP address that should be used for packet forwarding. The next hop is usually set to the IP address of the BGP router sending the update.

Establishing BGP Sessions

Overview

Understanding the [BGP](#) neighbor session establishment process is a key component to understanding the fundamental operation of the BGP protocol.

BGP is an [EGP](#) that has been designed for scalability and policy control. As a result, BGP requires neighboring routers to be explicitly configured before BGP routing updates can be sent between them. This situation differs from [IGPs](#) such as [EIGRP](#), [IS-IS](#), and [OSPF](#), that discover neighbors by using a broadcast packet or a hello protocol. In this lesson, BGP neighbor session establishment procedures are discussed.

Upon completing this lesson, you will be able to:

- Explain how BGP discovers neighbors
- Describe the BGP session establishment process
- Describe the role of the BGP keepalive in session establishment and maintenance
- Explain how optional MD5 authentication can protect sessions between BGP peers

BGP Neighbor Discovery

Unlike other routing protocols, [BGP](#) has no means of automatically detecting neighbors. The BGP protocol is carried in a [TCP](#) session, which must be opened from one router to the other. To do so, the router attempting to open the session must be manually configured with neighbor information indicating to which [IP address](#) to direct its connection attempts.

- BGP neighbors are not discovered; they must be configured manually.
- Configuration must be done on both sides of the connection.
- Both routers will attempt to connect to the other with a TCP session on port number 179.
- Only the session with the higher router-ID remains after the connection attempt.
- The source IP address of incoming connection attempts is verified against a list of configured neighbors.

The router that receives the incoming connection attempts does not answer them if the attempts are not from one of the configured neighbors. The [IP](#) source address of the connection attempt packet (TCP [SYN](#) packet) is verified against the list of IP addresses that the router itself would direct its connection attempts to.

To succeed in the connection attempts, both routers must be configured to reach each other. A side effect of this situation is that they will both attempt to connect. This side effect adds robustness to the session establishment process, but it also introduces the risk that two BGP sessions will be established between a pair of BGP routers.

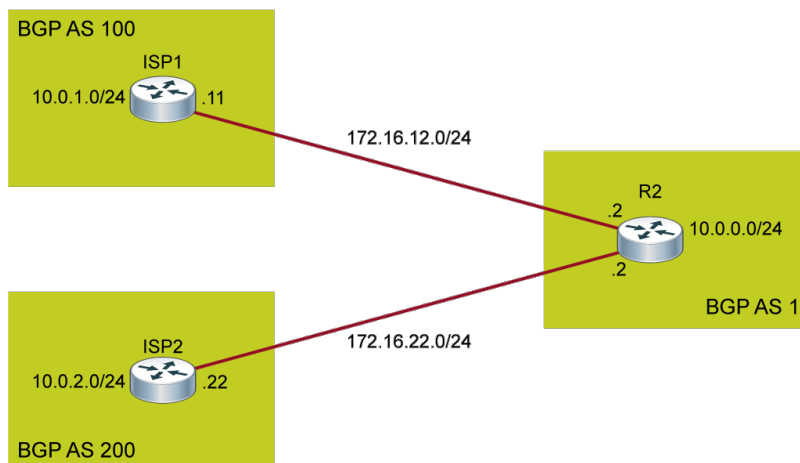
Two routers should have only a single BGP session between them. The router-ID values that are exchanged when the BGP session is established allow the BGP routers to detect when two parallel sessions exist. Only the session that was initiated by the router with the numerically higher router-ID will be retained. The other session is dropped.

A router may not open a BGP session to itself. If the configured neighbor IP address is, in fact, an IP address of the local router, the router recognizes the problem and tears down the session. The router-ID is also used for this verification.

BGP Neighbor Discovery Example

This example illustrates a small BGP network.

Small BGP Network



The network that is displayed in the figure serves as the sample network to generate printouts in the following examples.

Initially, all BGP sessions to the neighbors are idle.

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 31, main routing table version 31

Neighbor      V   AS  MsgRcvd MsgSent TblVer  InQ  OutQ  Up/Down  State/P
fxRcd
172.16.12.11  4   100      0      0       1    0    0  00:00:37 Idle
```

172.16.22.22	4	200	0	0	1	0	0	00:00:14	Idle

The **show ip bgp summary** command gives an overview of the BGP status. Each configured neighbor is listed in the output of the command. The IP address to which the connection attempts are directed is also displayed, along with the BGP version number, the remote [AS](#) number, some counter values, the status of the session, and how long ago the session changed state.

The "Idle" state indicates that the router is currently not attempting any connection establishments.

The various states for a BGP connection are Idle, Active, OpenSent, OpenConfirm, and Established.

Establishing a BGP Session

Before any connection attempt is made, the BGP peer relation must have left the Idle state and entered the Active state. For a BGP session between two routers in different autonomous systems, this status results when the IP address of the remote router becomes reachable on a directly connected interface.

- A TCP session is established when the neighbor becomes reachable.
- BGP Open messages are exchanged.

```
R2# debug ip tcp transactions
R2# debug ip bgp
R2# debug ip bgp events
BGP: 172.16.12.11 active went from Idle to Active
BGP: 172.16.12.11 open active, local address 172.16.12.2
TCP0: state was LISTEN -> SYNRCVD [179 -> 172.16.12.11(11374)]
TCP0: state was SYNRCVD -> ESTAB [179 -> 172.16.12.11(11374)]
TCBEFF01278 accepting EC3FFBE8 from 172.16.12.11.11374
BGP: 172.16.12.11 passive open to 172.16.12.2
BGP: 172.16.12.11 passive went from Idle to Connect
BGP: 172.16.12.11 passive rcv message type 1, length (excl. header) 38
BGP: ses global 172.16.12.11 (0xEFF0F1B8:0) pas Receive OPEN
BGP: 172.16.12.11 passive rcvd OPEN w/ remote AS 100, 4-byte remote AS 100
BGP: 172.16.12.11 passive went from Connect to OpenSent
BGP: 172.16.12.11 passive went from OpenSent to OpenConfirm
BGP: 172.16.12.11 passive went from OpenConfirm to Established
BGP(0): 172.16.12.11 was the first peer to be established for IPv4 Unicast
%BGP-5-ADJCHANGE: neighbor 172.16.12.11 Up
```

The debug output shows how the router creates a socket data structure and binds it to its local IP address 172.16.12.2 and the well-known port 179. Then the router sends a TCP SYN packet to the configured peer router IP address of 172.16.12.11 and a high port number 11374. The connection attempt succeeds, and the TCP session is then ready to transfer the BGP information.

The first BGP information that is sent is the BGP Open message. The BGP session then goes from the Active state to the OpenSent state while waiting for the other router to respond. If the peer router accepts the parameters in the Open message, it responds with its own Open message. When the local router receives this message, the state goes from OpenSent to OpenConfirm. The local router then verifies the peer router parameters in its Open message. If they are accepted, a keepalive packet is sent to signal this acceptance. The state is then "Established."

The BGP Open message contains the following:

- BGP version number
- AS number of the local router
- Holdtime
- BGP router identifier
- Optional parameters

The parameters in the BGP Open message are as follows:

- **Version number:** The suggested version number. The highest common version that both routers support is used. Most BGP implementations today use BGP4.
- **AS number:** The AS number of the local router. The peer router verifies this information. If it is not the AS number that is expected, the BGP session is torn down.
- **Holdtime:** The number of seconds that may elapse between reception of successive BGP messages. If the time is exceeded, the peer is considered dead. The two routers agree to use the lowest suggested value. When the session is established, both routers use keepalive messages to make sure that the hold timer does not expire. A suggested hold-timer value of 0 indicates that the timer never expires and no keepalives should be sent.
- **BGP identifier:** A number uniquely identifying the router. The Cisco router uses one of its IP addresses for this number, the router-ID. The router-ID is selected as the numerically highest IP address of any loopback interface. If there is no loopback interface, the router uses the highest IP address of any interface that is up at the time of the start of the BGP process.
- **Optional parameters:** TLV encoded. An example of optional parameters is session authentication.

BGP neighbors—steady state

- All neighbors shall be up (no state information).

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 16, main routing table version 16
15 network entries using 2220 bytes of memory
15 path entries using 960 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3500 total bytes of memory
BGP activity 78/63 prefixes, 84/69 paths, scan interval 60 secs

Neighbor      V    AS  MsgRcvd MsgSent TblVer  InQ  OutQ  Up/Down  State/P
fxRcd
172.16.12.11  4   100     18     15    16    0    0  00:11:46  6
172.16.22.22  4   200      5      8    16    0    0  00:00:27  9
```

When the BGP sessions are in the Established state, routing information exchange can take place. The **show ip bgp summary** command output here indicates that a session is established by not displaying any information at all in the "State" column.

The counter values show how many messages have been received and sent in the session. "InQ" shows how many messages have been received but not yet processed. A high InQ number indicates lack of [CPU](#) resources to process the input. "OutQ" shows how many outgoing messages are queued. A high OutQ number indicates lack of bandwidth to transmit the outgoing messages or CPU overload of the other router.

The BGP router uses "TblVer" (table version) to track the changes that need to be sent to the neighbors. There is a major table version number for the local BGP table. The table version number is displayed on the first line of output from this **show** command. There is also one table version number that is maintained for each of the neighbors of the BGP router; this number is displayed on the information line of the neighbors.

Whenever a BGP router enters a change into its BGP table, the major table version number is incremented and the changed route is tagged with this number. When the time comes to update a specific neighbor, the router scans the BGP table. All changes where the version number is between the neighbor version and the current table version are sent to the BGP neighbor in a single BGP routing update. After the entire table is scanned and all changes have been sent to the neighbor, the table version number of the neighbor is set to the highest value of the routes being sent.

A table version of a neighbor that is lower than the major table version indicates that the neighbor is not yet fully updated. The update interval for a neighbor in another AS is normally 30 seconds (the default value of the BGP advertisement timer).

In addition to the information about all sessions to all neighbors, the output also shows the amount of memory that is being used for the BGP data structures.

BGP Keepalives

TCP-based BGP sessions do not provide any means of verifying the presence of a BGP neighbor. After BGP has established the TCP session, the only method of verifying neighbor presence is to actually send BGP traffic. BGP traffic is sent over the TCP session with ACKs, and is therefore reliable. Successfully sending BGP traffic confirms the existence of a BGP neighbor.

- A TCP-based BGP session does not provide any means of verifying BGP neighbor presence:
 - Except when sending BGP traffic
- BGP needs an extra mechanism:
 - Keepalive BGP messages provide verification of neighbor existence.
 - Keepalive messages are sent every 60 seconds.

However, there are often long periods of time when no BGP traffic is sent between neighbors. During those periods, TCP implements no mechanism to check for the existence of the configured neighbor. BGP neighbors could therefore easily be disconnected during times of session inactivity. This situation would lead to incorrect routing information on the other side of the BGP session.

To avoid routing packets to a router that is no longer there, BGP needs an extra mechanism to make sure that a neighbor exists. BGP sends special keepalive messages during every keepalive interval to inform its peer of its presence. By default, this interval is every 60 seconds. If no BGP traffic is received within the selected holdtime interval, the BGP router sends a BGP notification message to the inactive peer and tears down their BGP session. The default BGP holdtime value is 180 seconds.

When changing the default values of keepalive and holdtime intervals, you must take care not to configure too big a keepalive interval in comparison to the holdtime. Too big a difference could result in resetting of the BGP session after only one keepalive message has been missed, making a network unstable. The suggested ratio of keepalive-to-holdtime interval is 1:3.

- Keepalive interval value is not communicated in the BGP Open message.
- Keepalive value is selected as follows:
 - Configured value, if local holdtime is used
 - Configured value, if holdtime of neighbor is used and $\text{keepalive} < (\text{holdtime} / 3)$
 - Smaller integer in relation to $(\text{holdtime} / 3)$, if holdtime of neighbor is used and $\text{keepalive} > (\text{holdtime} / 3)$

As opposed to the holdtime interval, BGP peers do not communicate the keepalive interval in the Open message. The selection of a keepalive interval is therefore based on the selected holdtime value. The selected holdtime value, which is used by both peers, is the smaller of both configured values.

The BGP process selects the keepalive interval value according to these conditions:

1. If the locally configured value of holdtime is selected (being the lower of two), the peers use the locally used keepalive interval.
2. If the holdtime interval of the neighbor is selected, and the locally configured keepalive interval is less than a third of the holdtime interval, the peers use the locally configured keepalive.
3. If the holdtime interval of the neighbor is selected, and the locally configured keepalive interval is more than a third of the holdtime interval, the peers use the smaller integer value in relation to $(\text{holdtime} / 3)$.

Keepalive Value Example

If the selected holdtime equals 17 seconds and the configured keepalive equals 10 seconds, the $(\text{holdtime} / 3)$ rule will be used to select the keepalive value. Therefore, $(17 / 3) = 5.67$, and the keepalive value that is used by BGP will equal 5 seconds.

MD5 Authentication

Authentication between [BGP](#) neighbors can be negotiated between BGP-speaking routers using optional parameters in the Open message.

- BGP peers may optionally use MD5 TCP authentication using a shared secret.
- Both routers must be configured with the same password (MD5 shared secret).
- Each TCP segment is verified.

If you are using [MD5](#) authentication, every [TCP](#) segment on the BGP session will be transmitted to the configured neighbor along with a checksum. The checksum is calculated together with a secret known by the two routers using the MD5 algorithm. The common secret is never transmitted on the network. If the receiver, which is using the same common secret, calculates the same checksum from the TCP segment, then the receiver can be pretty sure that the information is transmitted from the correct source and the information has not been altered.

Authentication of BGP sessions is a vital tool to avoid [DoS](#) attacks.



Cisco routers support key chain to configure BGP authentication. BGP supports only HMAC-MD5 and HMAC-SHA1-12 cryptographic algorithms.

Summary

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

- With interior routing protocols, adjacent routers are usually discovered through a dedicated hello protocol. In BGP, neighbors must be manually configured to increase routing protocol security.
- BGP neighbors, once configured, establish a TCP session and exchange the BGP Open message, which contains the parameters that each BGP router proposes to use.
- The router uses BGP keepalives to provide a verification of the existence of a configured BGP neighbor.
- MD5 authentication can be configured on a BGP session to help prevent spoofing, DoS attacks, or man-in-the-middle attacks.

Processing BGP Routes

Overview

Route processing is fundamental to the operation of [BGP](#). Knowledge of the BGP route selection process, route propagation, and how the BGP and IP routing tables are built is key to properly configuring BGP and troubleshooting BGP routing issues.

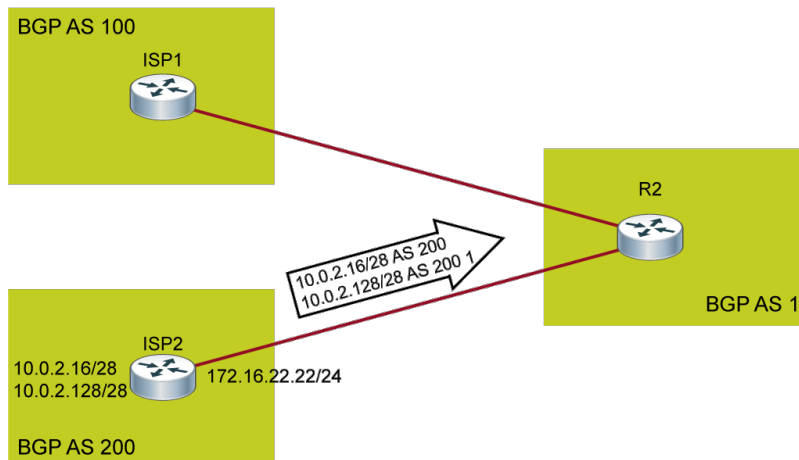
Upon completing this lesson, you will be able to:

- Describe BGP routing updates
- Explain how a router builds BGP tables
- Describe the route selection process in BGP
- Explain how a router propagates BGP routes to other BGP neighbors
- Explain how a router builds an IP routing table when it is using BGP
- Explain how BGP advertises local networks
- Describe the role of automatic summarization in BGP route processing

Receiving Routing Updates

After a [BGP](#) session is established, routing updates start to arrive. Each BGP routing update consists of one or more entries (routes). Each route is described according to the [IP address](#) and subnet mask, along with any number of attributes. The next-hop, [AS-path](#), and origin attributes must always be present. Other BGP attributes are optionally present.

Small BGP Network



The network in the figure serves as the sample network for generating printouts in the following examples.

Information from the BGP tables is exchanged after adjacency establishment.

```
R2# debug ip bgp updates
%BGP-5-ADJCHANGE: neighbor 172.16.22.22 Up
BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i,
metric 0, merged path 200, AS_PATH
BGP(0): 172.16.22.22 rcv UPDATE about 10.0.2.128/28 -- DENIED due to: AS-
PATH contains our own AS;
BGP(0): 172.16.22.22 rcvd 10.0.2.16/28
BGP(0): Revise route installing 1 of 1 routes for 10.0.2.16/28 -
> 172.16.22.22(global) to main IP table
```

The debug output shows how information about network 10.0.2.16/28 is received from neighbor 172.16.22.22. The neighbor indicates that [IP](#) packets to destination IP addresses in network 10.0.2.16/28 can be forwarded to the next-hop address 172.16.22.22. The AS path 200 indicates that the final destination is in AS 200. The metric is the [MED](#) value.

Network 10.0.2.128/28 is denied. The reason is AS path loop detection. The receiving router detects its own AS number in the AS path and silently discards (denies) the route.

Building BGP Table

All routes that are received from a neighbor are saved in the router memory. Therefore, there is no need to retransmit or refresh any unchanged information.

All inbound updates are placed into the BGP table.

```
R2# show ip bgp
BGP table version is 11, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*  1.0.0.0/24      172.16.22.22         0           0 200 99 i
*> 1.0.0.0/24      172.16.12.11         0           0 100 99 i
*> 10.0.1.0/24     172.16.12.11         0           0 100 i
*> 10.0.2.0/28     172.16.22.22         0           0 200 i
*> 10.0.2.16/28    172.16.22.22         0           0 200 i
*> 10.0.2.32/28    172.16.22.22         0           0 200 i
*> 10.0.2.48/28    172.16.22.22         0           0 200 i
*> 10.0.2.64/28    172.16.22.22         0           0 200 i
*> 10.0.2.80/28    172.16.22.22         0           0 200 i
*> 10.0.2.96/28    172.16.22.22         0           0 200 i
*> 10.0.2.112/28   172.16.22.22         0           0 200 i
```

When there is more than one way to reach a particular network, the local router selects one of them as the best. The best alternative might be later lost because the neighboring router withdraws the route or the neighboring router is no longer reachable. However, the remaining alternatives are still stored in memory and a new alternative is selected as the best without involving other [BGP](#) routers.

The **show ip bgp** router command gives an overview of all routing information that is received from all neighbors. The command displays basic information about each route on a single line. The output is sorted—alternatives to reach the same network are displayed on consecutive lines. The network number is displayed only on the first lines indicating the same network. The network column is left blank on the consecutive lines indicating alternatives to reach the same network.

The router selects only one of the alternatives as the best path toward the destination. This alternative is indicated with the ">" sign.

BGP Route Selection Criteria

When a router has more than one alternative route to reach the same [IP](#) subnet (network and mask), the router has to select one of the routes as best in its default mode of operation. To make this selection, the router uses the [BGP](#) attributes that are attached to the various updates.

- Exclude routes with inaccessible next hop
- Prefer highest weight (local to router)
- Prefer highest local preference (global within AS)
- Prefer routes that the router originated
- Prefer shortest AS path (only length is compared)
- Prefer lowest origin code (IGP < EGP < Incomplete)
- Prefer lowest MED
- Prefer external (EBGP) paths over internal (IBGP)
- For IBGP paths, prefer path through closest IGP neighbor
- For EBGP paths, prefer oldest (most stable) path
- Prefer paths from router with the lowest BGP router-ID

The selection criteria are checked in the order that is indicated in the following steps. The first check that indicates a difference is used for route selection, and no further testing is done.

1. The router checks whether the next-hop attribute indicates an [IP address](#) that is reachable according to the current routing table. It is not necessary to have a direct connection to the next hop. It can very well be several router hops away and the route to it learned by the [IGP](#). If the next hop is not reachable, the router does not consider the BGP route as a candidate to become selected as the best.
2. The router prefers the route with the higher weight. The weight is not carried with the updates; it is a value that is assigned to the route by the local router and considered only within the router itself.
3. If the local preference attributes are different, the route with the highest value is selected as best.
4. If one of the routes is injected into the BGP table by the local router, the local router prefers it to any routes that it receives from other BGP routers.
5. At this point, the lengths of the [AS](#) paths are compared (the content is not checked; only the number of autonomous systems in each AS path is counted). The route with the shortest length is selected.
6. If the AS-path lengths are the same, the origin code is checked. BGP will prefer the path with the lowest origin type: IGP is lower than [EGP](#), and EGP is lower than Incomplete.
7. The router next compares [MED](#) values but only if it receives the updates from the same neighboring AS. Routes with a lower MED are preferred.
8. At this point, it is clear that the destination network is outside the local AS and that there is not much difference among the alternatives. Because the IP packets to the destination network must leave the AS, it is better that they do so as quickly as possible. If any of the alternatives are received from a BGP peer in another AS, that alternative is preferred.
9. If the router receives all alternatives from peer routers in the local AS, each of them will indicate an exit point, and the closest exit is used. Distance to the exit point is calculated by comparing the IGP costs against the BGP next hops, as indicated in the routing table.
10. If the router receives all alternatives from [EBGP](#) neighbors, the most stable path (the oldest path) is preferred.
11. If the router still cannot differentiate among the routes, it nevertheless has to make a decision and select the best route. It checks the BGP sessions on which it received the updates and chooses the route that was received on the session for which the peer router has the lowest BGP router-ID.

The router makes the final test only after it has made all other checks and determined that all alternative routes are equally good.

BGP Route Propagation

A local router propagates only the route that it selected as best to the neighbors. However, the router never sends a route back on the same [BGP](#) session upon which it was received. On the contrary, when it selects a neighbor as the best next hop, the router makes sure that the neighbor is not pointing back to the local router. The router accomplishes this task by "poisoning" the route (marking the route unreachable) and sending a withdraw message to that neighbor.

The best BGP routes are propagated to BGP neighbors.

```
R2# debug ip bgp updates
BGP(0): (base) 172.16.12.11 send UPDATE (format) 10.0.2.0/28, next 172.16.12.2, metric 0, path 200
BGP(0): 172.16.12.11 NEXT_HOP is on same subnet as the bgp peer and set to 172.16.12.11 for net 10.0.1.0/24, flags 200, sb: AC100C00, mask: FFFFFFF0
BGP(0): (base) 172.16.12.11 send UPDATE (format) 10.0.1.0/24, next 172.16.12.11, metric 0, path 100
BGP(0): (base) 172.16.12.11 send UPDATE (format) 10.0.0.0/28, next 172.16.12.2, metric 0, path Local
```

The router conducts route poisoning to avoid a potential routing loop problem in which a neighbor router selected as the best next hop might rely on the local router as the best next hop.

The process of preventing routing information from being sent back to the source of information is called "split horizon."

Building IP Routing Table

The route in the [BGP](#) table that BGP selects as the best is a candidate for installation in the [IP](#) forwarding table or routing table.

The best BGP routes are copied into the IP routing table based on administrative distance.

```
R2# show ip route
<... output omitted ...>

Gateway of last resort is not set

  1.0.0.0/24 is subnetted, 1 subnets
B       1.0.0.0 [20/0] via 172.16.22.22, 00:03:07
      10.0.0.0/8 is variably subnetted, 20 subnets, 3 masks
<... output omitted ...>
B       10.0.1.0/24 [20/0] via 172.16.12.11, 00:03:06
B       10.0.2.0/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.16/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.32/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.48/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.64/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.80/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.96/28 [20/0] via 172.16.22.22, 00:06:46
B       10.0.2.112/28 [20/0] via 172.16.22.22, 00:06:46
<... output omitted ...>
```

Before a route can be installed, the router has to check whether there is any other routing protocol that has information about the same subnet (network and mask). If the subnet is known via different sources, the router uses the [AD](#) to determine which source to use. AD is a rating of the trustworthiness of a routing information source. AD is often expressed as a numerical value between 0 and 255. The higher the value, the lower the trustworthiness rating. In this case, the router will install the route with the lowest AD.

The output from the **show ip route** command indicates which routes in the routing table were installed using the BGP information. Those routes are denoted with the letter "B." The AD is shown in the command output as the first number within the brackets.

In this example, network 1.0.0.0/24 is reachable via 172.16.22.22. After the router has installed the route in the routing table, user data traffic starts to be forwarded.

Advertising Local Networks

The [BGP](#) routing process can inject new routes into the BGP table. A router will propagate newly injected routes to neighboring BGP peers if it selects them as best. This way the router provides the neighboring autonomous systems with information about networks that are reachable in the local [AS](#). This process is called advertising, originating, or announcing local routes.

- The BGP router process keeps a list of local networks (defined with the **network** command or through redistribution).
- The BGP process periodically scans the IP routing table and inserts or revokes routes from the BGP routing table based on their presence in the IP routing table.

The BGP process can inject local routes in two different ways:

- A list of networks is configured on the router under the BGP router process using the **network** configuration command. The listed networks are candidates for being injected. Networks are injected only if they appear in the routing table. In the case where the [IGP](#) that is used within the AS finds a valid path to them, the routes will be in the routing table.
- Routes that are learned by another routing protocol are redistributed. The IGP that is used with the AS can also act as a source of routing information about local networks.

Advertising Local Networks Example

In this example, network 10.0.0.0/28 is directly connected to interface Loopback1.

The BGP route is revoked after the network is removed from the routing table.

```
R2# debug ip routing
R2# debug ip bgp updates
is_up: Loopback1 0 state: 6 sub state: 1 line: 0
RT: interface Loopback1 removed from routing table
RT: del 10.0.0.0 via 0.0.0.0, connected metric [0/0]
RT: delete subnet route to 10.0.0.0/28
RT: del 10.0.0.1 via 0.0.0.0, connected metric [0/0]
RT: delete subnet route to 10.0.0.1/32
BGP(0): route 10.0.0.0/28 down
BGP(0): no valid path for 10.0.0.0/28
BGP: topo global:IPv4 Unicast:base Remove_fwdroute for 10.0.0.0/28
BGP(0): (base) 172.16.12.11 send unreachable (format) 10.0.0.0/28
```

The route to 10.0.0.0/28 was previously installed in the BGP table because it was listed with a network statement and it was in the routing table as directly connected. When the Loopback1 interface goes down, the router removes the directly connected route from its routing table. Because the route no longer exists in the routing table, it must also be removed from the BGP table.

Because there has been a change in the BGP table, the BGP neighbors must be informed. The router sends a BGP update message to its neighbor indicating that network 10.0.0.0/28 is now unreachable.

The BGP route is advertised after the network appears in the routing table.

```
RT: add 10.0.0.0/28 via 0.0.0.0, connected metric [0/0]
RT: interface Loopback1 added to routing table
RT: updating connected 10.0.0.1/32 (0x0):
    via 0.0.0.0 Lo1
RT: add 10.0.0.1/32 via 0.0.0.0, connected metric [0/0]
BGP(0): route 10.0.0.0/28 up
BGP: topo global:IPv4 Unicast:base Remove_fwdroute for 10.0.0.0/28
BGP(0): redistributedlocal route 10.0.0.0/28 modified
BGP(0): (base) 172.16.12.11 send UPDATE (format) 10.0.0.0/28, next 172.16.12.2, metric 0, path Local
```

In this example, network 10.0.0.0/28 is listed with a network statement in the BGP process. However, the network was not in the routing table of the router, so the network was not injected into its BGP table.

Later, the Loopback1 interface comes back up again. This reappearance means that the network 10.0.0.0/28 is now in the routing table as a directly connected route. As a result, the router once again injects the 10.0.0.0/28 network into its BGP table and

then updates its configured neighbor.

Automatic Summarization

When a **BGP** router is configured to locally announce routes into BGP, the behavior of the **network** command varies depending on whether automatic summarization is enabled or disabled. When automatic summarization is enabled, BGP summarizes the locally originated BGP networks (network x.x.x.x) to their classful boundaries.

Verify if BGP automatic summarization is disabled or enabled. The following examples will have BGP automatic summarization enabled.

- Verify BGP automatic summarization.
- Enable automatic summarization when:
 - Summarization of IGP-to-BGP redistributed routes to major network boundary required
 - Using classful **network** command to summarize subnets to a major network boundary
- Disable automatic summarization when:
 - Summarization on IGP-to-BGP redistribution not desired
 - Using classless variant of the **network** command

When a subnet exists in the routing table and the following three conditions are satisfied, then any subnet (component route) of that classful network in the local routing table will prompt BGP to install the classful network into the BGP table:

- A classful network statement for that network exists in the routing table.
- A classful mask has been configured on that network statement.
- Automatic summarization is enabled.

When automatic summarization is disabled, the routes that are introduced locally into the BGP table are not summarized to their classful boundaries.

The behavior of the redistribution procedure in BGP is also influenced by the configuration of automatic summarization on the router. When enabled, all redistributed subnets will be summarized to their classful boundaries in the BGP table. When disabled, all redistributed subnets will be present in their original form in the BGP table.

Enable automatic summarization in BGP when the summarization of subnets to their classful boundaries will not introduce flawed information into the BGP table. In other words, leave automatic summarization enabled only when you are using a fully assigned classful network matching the network that was summarized in BGP.

Whenever possible, use the classless variant of the **network** command, specifying the subnet mask length of the network. When you are redistributing networks into BGP, the preferred method is to disable automatic summarization. Disabling automatic summarization ensures that correct information is inserted into the BGP table of the router.

Automatic Summarization Example

In this example, five subnets of the major class A network 10.0.0.0/8 (10.0.0.0/28, 10.0.0.16/28, 10.0.0.32/28, 10.0.0.48/28, and 10.0.0.64/28) exist in the routing table.

```
R2# show ip route
      10.0.0.0/8 is variably subnetted, 11 subnets, 3 masks
<... output omitted ...>
C       10.0.0.0/28 is directly connected, Loopback1
C       10.0.0.16/28 is directly connected, Loopback2
C       10.0.0.32/28 is directly connected, Loopback3
C       10.0.0.48/28 is directly connected, Loopback4
C       10.0.0.64/28 is directly connected, Loopback5
<... output omitted ...>
```

- Five subnets for 10.0.0.0 exist in the routing table.
- Automatic summarization is enabled for BGP.
- BGP has been configured to locally announce 10.0.0.0.

```
router bgp 1
 network 10.0.0.0
 auto-summary
```

```

R2# show ip bgp
BGP table version is 12, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  1.0.0.0/24       172.16.12.11              0           0 100 99 i
*>  10.0.0.0         0.0.0.0                   0           32768 i
*>  10.0.1.0/24      172.16.12.11              0           0 100 i

```

- Classful network summary is inserted into BGP table.

When you are inserting networks into the BGP table with the classful **network** command and automatic summarization is disabled, no insertion into the BGP table will occur unless an exact match exists in the IP routing table (meaning that a classful network has to be present in the [IP](#) routing table).

When automatic summarization is enabled, the major **network** command will summarize all subnets in the IP routing table to their major network boundary.

There is a classful **network** command, and automatic summarization is enabled for BGP. This setup results in the insertion of a classful network summary into the BGP table, instead of separate subnets.

Subnets 10.0.0.0/28, 10.0.0.16/28, 10.0.0.32/28, 10.0.0.48/28, and 10.0.0.64/28 were summarized during insertion into the BGP table to the classful network 10.0.0.0/8. This action occurred because a classful **network** command and automatic summarization were configured on the router. If automatic summarization is disabled, no insertion into the BGP table occurs at all.

The locally sourced summary has all the attributes of a locally sourced BGP route (next hop = 0.0.0.0, weight = 32768, empty [AS](#)-path list), and is marked as having an [IGP](#) origin (being sourced with the **network** command).

```

R2# show ip route
<... output omitted ...>
  10.0.0.0/8 is variably subnetted, 11 subnets, 3 masks
C       10.0.0.0/28 is directly connected, Loopback1
C       10.0.0.16/28 is directly connected, Loopback2
C       10.0.0.32/28 is directly connected, Loopback3
C       10.0.0.48/28 is directly connected, Loopback4
C       10.0.0.64/28 is directly connected, Loopback5
<... output omitted ...>

```

- Five subnets 10.0.0.0 exist in the routing table.
- Automatic summarization is enabled for BGP.
- BGP has been configured to redistribute connected routes into BGP.

```

router bgp 1
 redistribute connected
 auto-summary

```

```

R2# show ip bgp
BGP table version is 15, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  1.0.0.0/24       172.16.12.11              0           0 100 99 i
*>  10.0.0.0         0.0.0.0                   0           32768 ?
*>  10.0.1.0/24      172.16.12.11              0           0 100 i
*>  172.16.0.0       0.0.0.0                   0           32768 ?

```

- Classful network summary is inserted into BGP table.

In this example, automatic summarization is enabled, resulting in the summarization of redistributed subnets to their classful boundaries. Subnets 10.0.0.0/28, 10.0.0.16/28, 10.0.0.32/28, 10.0.0.48/28, and 10.0.0.64/28 will be summarized into the single class A network 10.0.0.0/8. The network 10.0.0.0/8 is a locally sourced summary with all the attributes of a locally sourced BGP route (next hop = 0.0.0.0, weight = 32768, empty AS-path list). The origin of the route is marked as incomplete because the route is sourced through redistribution.

If automatic summarization was disabled, more specific routes would be present in the BGP table instead of the summary prefix 10.0.0.0/8.

Summary

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

- After BGP sessions are established between BGP routers, they can start exchanging routing updates.
- All updates that are received from BGP neighbors are stored in the BGP table, regardless of whether they are used.
- The route selection process takes into account various BGP attributes that are attached to the route, as well as local decisions (indicated with weights).
- Only the best BGP routes are propagated to other BGP routers.
- Only the best BGP routes are installed in the local IP routing table.
- Every BGP router can also originate the routes in BGP. The routes to be originated are entered manually in the BGP routing process or redistributed into BGP from an IGP.

Configuring Basic BGP

Overview

Basic [BGP](#) configuration is critical to any successful BGP implementation. Network administrators use Cisco IOS commands that are included in this lesson in all BGP implementations. Thorough knowledge of the commands in this lesson is therefore crucial to ensuring a successful implementation using BGP.

Upon completing this lesson, you will be able to:

- Identify the Cisco IOS command that is required to configure the BGP routing process
- Identify the Cisco IOS commands that are required to configure external BGP neighbors
- Identify the Cisco IOS commands that are required to configure basic timers and MD5 authentication in BGP
- Identify the commands that are required to announce local networks in BGP
- Describe BGP route redistribution, including the commands that are required to configure BGP route redistribution
- Describe the classless behavior of BGP and identify the Cisco IOS command that is required to configure BGP for classless operation
- Describe BGP route aggregation, including the Cisco IOS commands that are required to configure basic BGP route aggregation
- Describe the BGP Conditional Route Injection feature
- Describe the BGP Support for TTL Security Check feature
- Determine when BGP route aggregation is not appropriate in multihomed topologies

BGP Routing Process

You will use the **router bgp** command to start the [BGP](#) routing process in the router. There can be, at most, one BGP process in a router. It must be assigned the local [AS](#) number.

```
router(config)# router bgp as-number
```

- Starts BGP routing.
- Get your AS number from American Registry for Internet Numbers (<http://www.arin.net>) or Réseaux IP Européens (<http://www.ripe.net>).
- Use private AS numbers (between 64,512 and 65,534 or between 4,200,000,000 and 4,294,967,294) if you run BGP in a private network.
- Only one BGP routing process per router is allowed.

The AS number is a 16-bit or 32-bit integer. New 32-bit AS numbers were created when the AS number pool from [IANA](#) approached exhaustion. The AS number pool was extended from 16 to 32 bits. It must uniquely identify the AS among all routers that are exchanging BGP routing information, either directly or indirectly. This requirement means that the AS numbers must be unique when BGP information is exchanged with the Internet.

The AS number can be a public AS number or a private AS number. Public AS numbers range from 1 to 64,534 or 131,072 to 4,199,999,999. An Internet registry ([ARIN](#): <http://www.arin.net> or [RIPE](#): <http://www.ripe.net>) assigns public AS numbers. Private AS numbers range from 64,512 to 65,534 or 4,200,000,000—4,294,967,294. Private AS numbers are never propagated onto the public Internet.

Configuring External Neighbors

BGP does not automatically discover neighbors. You will need to explicitly configure BGP neighbors. The local router will try to connect to the indicated [IP address](#) and also accept incoming connection attempts from the indicated IP address.

```
router(config-router)# neighbor ip-address remote-as as-number
```

- Defines an external neighbor
- External neighbor has to be reachable over directly connected subnet

```
router(config-router)# neighbor ip-address description neighbor description
```

- Assigns a description to an external neighbor.

```
router(config-router)# neighbor ip-address shutdown
```

- Disables a BGP neighbor

The first attribute that you must configure with a new neighbor is the remote [AS](#) number in which the neighbor is taking part. When the [TCP](#) session is established between BGP routers, each router verifies the configured remote AS number with the exchange of BGP Open messages.

You may optionally configure other attributes with the neighbor on successive configuration lines, referring to the same neighbor IP address but indicating different attributes.

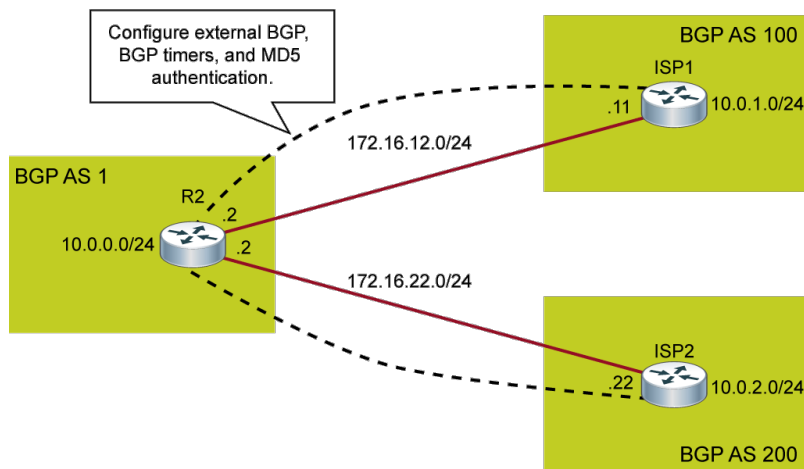
With the **neighbor description** command, a description (text string) can be entered that describes the neighbor.

When you are debugging or troubleshooting, you can use the **neighbor shutdown** command to temporarily disable BGP neighbor without deleting BGP neighbor configuration. You can shut down the BGP neighbor during extensive modification of routing policies to prevent inconsistent routing data.

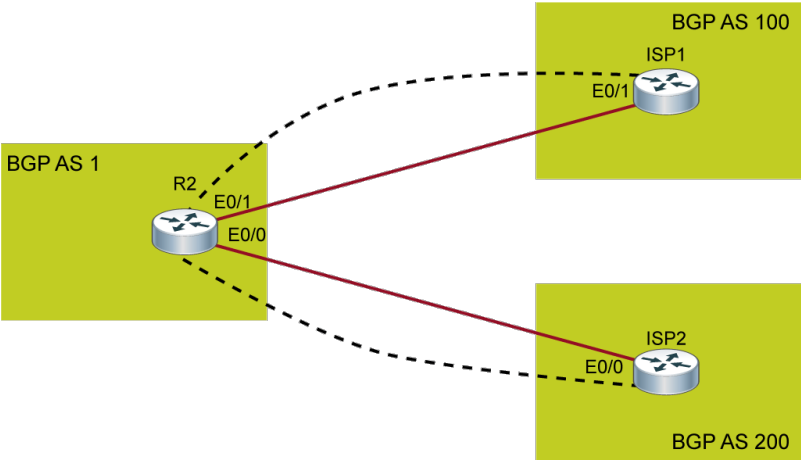
Discovery 1: Configure Basic BGP

Overview

Through this discovery, you will learn how to configure external [BGP](#), BGP timers, and [MD5](#) authentication.



Topology



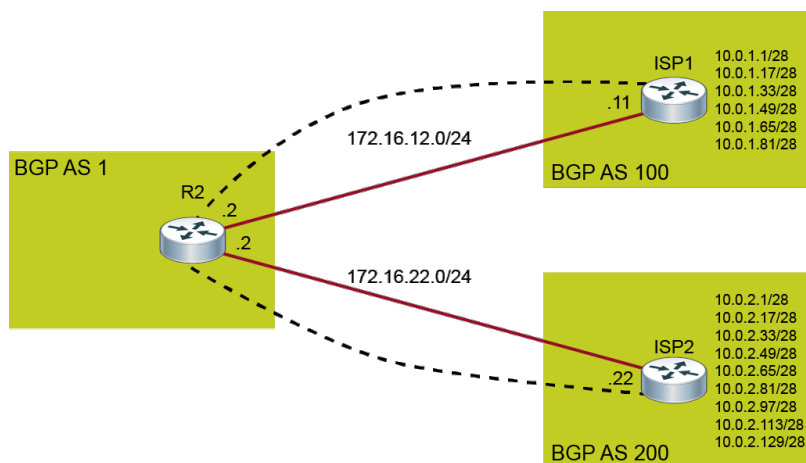
Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks



Configure Basic BGP

Discovery Steps

Step 1

You will configure an external BGP session between R2 and ISP1 routers. On the R2 router, configure BGP external neighbor ISP1.

ISP1 has IP address 172.16.12.11 and BGP AS 100. R2 router is already configured in the BGP AS 1.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 remote-as 100
```

Step 2

On the R2 router, verify the state of the BGP session.

To verify the BGP session, use the **show ip bgp summary** command.

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 34, main routing table version 34
9 network entries using 1332 bytes of memory
9 path entries using 576 bytes of memory
1/1 BGP path/bestpath attribute entries using 136 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2068 total bytes of memory
BGP activity 15/6 prefixes, 21/12 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.12.11    4   100      0       0        1    0    0 never      Idle
172.16.22.22    4   200    1359    1357       34    0    0 20:29:01      9
```

You should see "Idle" state next to the neighbor 172.16.12.11.

Step 3

On the ISP1 router, configure BGP external neighbor R2.

R2 has IP address 172.16.12.2 and BGP AS 1. ISP1 router is already configured in the BGP AS 100.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.12.2 remote-as 1
```

Shortly, after configuring external BGP neighbor on the ISP1 router, you will see the following message on the ISP1 router's console:

```
*Feb  5 01:13:18.858: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Up
```

External BGP session between R2 and ISP1 is successfully established.

Step 4

On the R2 router, verify the state of the BGP session.

To verify the BGP session, use the **show ip bgp summary** command.

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 40, main routing table version 40
15 network entries using 2220 bytes of memory
15 path entries using 960 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3500 total bytes of memory
BGP activity 21/6 prefixes, 27/12 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
```

172.16.12.11	4	100	32	36	40	0	0	00:25:36	6
172.16.22.22	4	200	1398	1398	40	0	0	21:04:31	9

Instead of state information, you should see the number of prefixes router received from its neighbor. Now you know that the state is Established and that router R2 received 6 prefixes from the ISP1 router.

Step 5

On the R2 router confirm that BGP state is established.

To confirm BGP state, use the **show ip bgp neighbor** command.

```
R2# show ip bgp neighbors 172.16.12.11
BGP neighbor is 172.16.12.11, remote AS 100, external link
BGP version 4, remote router ID 10.0.1.81
BGP state = Established, up for 00:32:50
Last read 00:00:46, last write 00:00:11, hold time is 180, keepalive interval is 60 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised and received
  Multisession Capability:
  Stateful switchover support enabled: NO for session 1
```

From the output, also observe the default hold time and keepalive interval.

Configuring BGP Timers

Changing the BGP default holdtime and keepalive timers is usually not recommended. The defaults (keepalive: 60 seconds; holdtime: 180 seconds) should work fine in most situations. If for any reason a faster BGP response to a peer down event is needed, the neighbor timers on the router can be reduced. For example, in scenarios where multiple paths toward destinations are available. This reduction will result in a faster detection of a lost peer and faster switching to the alternate path in the BGP table, thus improving convergence.

```
router(config-router)# timers bgp keepalive holdtime
```

- Changes the default values of BGP timers per BGP process.
- Only the holdtime value is communicated in the BGP Open message.
- Both peers use the smallest configured holdtime value on BGP peers.

```
router(config-router)# neighbor ip-address timers keepalive holdtime
```

- Changes the default values of BGP timers per specific neighbor.
- Overrides the **bgp** settings of the timers.

Suppose that there is a BGP router with an expired holdtime, which means that no BGP traffic was received within the holdtime interval. This BGP router will send a notification to its BGP peer, notifying it as to the reason for closing the session. The BGP router on which the holdtime has expired moves the inactive peer into the Idle state. After a certain time interval, a BGP router again tries to reconnect to the previously disconnected BGP peer and will also accept connection attempts from that peer. This time interval is determined by auto-enable and connection timers.

To set the timers for a specific BGP peer or peer group, use the **neighbor timers** router configuration command. This command overrides the values that you have set using the **timers bgp** command.

Step 6

You will change BGP timers for session between R2 and ISP1 routers. On the R2 router, change the default hold time and keepalive interval for ISP1 neighbor only.

Use a hold time of 30 seconds and keepalive interval 10 seconds.


```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 timers 10 30
```

Step 7

On the ISP1 router, change the default hold time and keepalive interval for BGP process.
Use hold time of 30 seconds and keepalive interval 10 seconds.

```
ISP1(config)# router bgp 100
ISP1(config-router)# timers bgp 10 30
```

Timers are negotiated at BGP neighbor establishment. On the ISP1 router, clear BGP sessions by using the following command:

```
ISP1# clear ip bgp *
*Feb  5 02:41:15.321: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Down User reset
*Feb  5 02:41:15.321: %BGP_SESSION-5-
ADJCHANGE: neighbor 172.16.12.2 IPv4 Unicast topology base removed from session User reset
*Feb  5 02:41:15.382: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Up
```

Step 8

On the R2 router, verify the BGP hold time and keepalive interval.
BGP timers have changed for ISP1 neighbor:

```
R2# show ip bgp neighbors 172.16.12.11
BGP neighbor is 172.16.12.11, remote AS 100, external link
  BGP version 4, remote router ID 10.0.1.81
  BGP state = Established, up for 00:01:03
  Last read 00:00:08, last write 00:00:00, hold time is 30, keepalive interval is 10 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```

BGP timers are default for ISP2 neighbor:

```
R2# show ip bgp neighbors 172.16.22.22
BGP neighbor is 172.16.22.22, remote AS 200, external link
  BGP version 4, remote router ID 10.0.2.129
  BGP state = Established, up for 00:11:01
  Last read 00:00:57, last write 00:00:06, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```

Configuring MD5 Authentication

To enable MD5 authentication on a TCP connection between two BGP peers, use the **neighbor password** router configuration command.

```
router(config-router)# neighbor ip-address password string
```

- Enables MD5 authentication on a specific BGP session.
- Password string on both routers must match.

Step 9

You will enable MD5 authentication for BGP session between R2 and ISP1 routers. On the R2 router, enable MD5 authentication on the BGP session to the ISP1 neighbor.

Use MD5 authentication password **cisco**.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 password cisco
```

Step 10

On the ISP1 router, enable MD5 authentication on the BGP session to the R2 neighbor.
Use MD5 authentication password **cisco**.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.12.2 password cisco
```

Make sure the password that is used on ISP1 is same as password used on R2. Space character after password will become part of the password.

If password is misconfigured, you will see the following message on the router console:

```
%TCP-6-
BADAUTH: Invalid MD5 digest from 172.16.12.2(60123) to 172.16.12.11(179) (RST) tableid - 0
```

Step 11

On the ISP1 router, verify that BGP session to the R2 router is established.

```
ISP1# show ip bgp neighbors 172.16.12.2
BGP neighbor is 172.16.12.2, remote AS 1, external link
BGP version 4, remote router ID 172.16.22.2
BGP state = Established, up for 00:08:19
Last read 00:00:02, last write 00:00:02, hold time is 30, keepalive interval is 10 seconds
Configured hold time is 30, keepalive interval is 10 seconds
Minimum holdtime from neighbor is 0 seconds
Neighbor sessions:
  1 active, is not multisession capable (disabled)
Neighbor capabilities:
  Route refresh: advertised and received(new)
  Four-octets ASN Capability: advertised and received
  Address family IPv4 Unicast: advertised and received
  Enhanced Refresh Capability: advertised and received
```

Step 12

On the R2 router, verify that BGP prefixes are received from ISP1 router.
You should see six BGP prefixes received from ISP1 router.

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 28, main routing table version 28
15 network entries using 2220 bytes of memory
15 path entries using 960 bytes of memory
2/2 BGP path/bestpath attribute entries using 272 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3500 total bytes of memory
BGP activity 63/48 prefixes, 75/60 paths, scan interval 60 secs

Neighbor      V      AS MsgRcvd MsgSent   TblVer   InQ  OutQ  Up/Down  State/PfxRcd
172.16.12.11  4      100     69     73       28    0    0 00:10:15        6
172.16.22.22  4      200     19     24       28    0    0 00:13:06        9
```

Announcing Networks in BGP

Before you inject any local routing information into [BGP](#) table for advertising to other BGP routers, some basic configuration is required.

Only administratively defined networks are announced in BGP.

- Manually configure networks to be announced.
- Use redistribution from IGP.
- Use aggregation to announce summary prefixes.

There are two ways to do this configuration:

- List the network numbers that are candidates to be advertised using the **network** configuration command. If any of the listed networks are reachable by the local router, according to its routing table, then the network is injected as a route into the BGP table.
- Redistribute routing information that has been learned by other routing protocols into the BGP table. You can use the [IGP](#) that is used within the [AS](#). Any route that is known by the local IGP can be injected into the BGP table using route redistribution between the IGP and BGP on the local router.

A router can also introduce new routing information into the BGP table by summarizing routes already there. This activity is called route aggregation and also requires configuration.

Any route that the router introduces into the BGP table will appear as a new route. The AS-path attribute for such a route will be empty, indicating a local route. The AS path changes later as the route passes AS boundaries.

```
router(config-router)# (no) auto-summary
```

- Enables or disables summarization of networks before insertion into the BGP table:
 - Locally inserted networks (using the **network** command)
 - Redistributed routes
- Enabled by default

When the router is configured to locally announce routes into BGP, the behavior of the **network** command varies depending on whether automatic summarization is enabled or disabled. When automatic summarization is enabled, the command summarizes locally originated BGP networks to their classful boundaries. By default, automatic summarization is enabled for BGP.

When a subnet exists in the routing table and the following three conditions are satisfied, then any subnet (component route) of that classful network in the local routing table will prompt BGP to install the classful network into the BGP table:

- A classful network statement for that network exists in the routing table.
- A classful mask has been configured on that network statement.
- Automatic summarization is enabled.

When automatic summarization is disabled, the routes that are introduced locally into the BGP table are not summarized to their classful boundaries.

The BGP **auto-summary** command is also responsible for the behavior of the redistribution procedure in BGP. When the command is enabled, all redistributed subnets will be summarized to their classful boundaries in the BGP table. When it is disabled, all redistributed subnets will be present in their original form in the BGP table.

```
router(config-router)# network major-network-number
```

To manually define a major network

- Allows advertising of major networks into BGP.
- At least one of the subnets must be present in the routing table.
- Behavior depends on the presence of the **auto-summary** command.
- The meaning of the **network** command in BGP is completely different from any other routing protocol.

To specify the networks to be advertised by the BGP routing process, use the **network** router configuration command. To remove an entry, use the **no** form of this command.



The meaning of the **network** command in BGP is radically different from the meaning of the command in other routing protocols. In all other routing protocols, the **network** command indicates interfaces over which the routing protocol will be run. In BGP, it indicates only which routes should be injected into the BGP table on the local router. Also, BGP never runs over individual interfaces; it is run over [TCP](#) sessions with manually configured neighbors.

The **network** command with no **mask** option uses the classful approach to insert a major network into the BGP table. At least one subnet of the specified major network needs to be present in the [IP](#) routing table to allow BGP to start announcing the major network as a BGP route. If automatic summarization is disabled, an exact match is required.

```
router(config-router)# network major-network-number route-map route-map-name
```

- The addition of the **route-map** option allows network parameters to be modified before you enter them into the BGP table.
- The **route-map** option can be used for the following:
 - Changing the weight value of a locally sourced route
 - Tagging sourced routes with BGP communities
 - Setting the local preference for a specific network
 - Changing the value of the MED for a specific network

When the router is configured to insert routes into the BGP table, the default attributes of locally sourced routes can be modified with the inclusion of the **route-map** option in the basic **network** command.

The attached route-map can change the following attributes of locally sourced networks with the **network** command:

- **Weight (default value = 32768):** The weight attribute is a special Cisco attribute that is used in the path selection process when there is more than one route to the same destination. Because weight is considered before local preference in BGP route selection, locally sourced routes are always preferred, unless the weight value is modified.
- **Community (default value = nonexistent):** Used for tagging routes at their source.
- **Local preference (default value = 100):** Used for AS-wide BGP best-path selection.
- **MED (default value = 0):** Used for return-path selection in topologies where multiple exit points to the same neighbor AS exist.

Announcing Networks in BGP Example

If a subnet existing in the routing table is 75.75.75.0/24, and network 75.0.0.0 is configured under the **router bgp** command (assuming that automatic summarization is enabled), BGP will introduce the classful network 75.0.0.0/8 in the BGP table. If the following three conditions are not all met, then BGP will not install any entry in the BGP table unless there is an exact match in the IP routing table:

- A classful network statement for the network exists in the routing table.
- A classful mask has been configured on that network statement.
- Automatic summarization is enabled.

Redistributing Routes into BGP

There are two alternatives for injecting local routes into the [BGP](#) table: list them using the **network** command or **redistribute** them. Listing the routes gives you total control over networks that could possibly be advertised by BGP. This option is very desirable for multihomed customers or [ISPs](#). On the other hand, this approach requires many configuration commands that could be difficult to maintain.

- Easier than listing networks in BGP process in large networks.
- Redistributed routes carry origin attribute "incomplete."
- Always filter redistributed routes to prevent route leaking.
- Avoid in service provider environments.

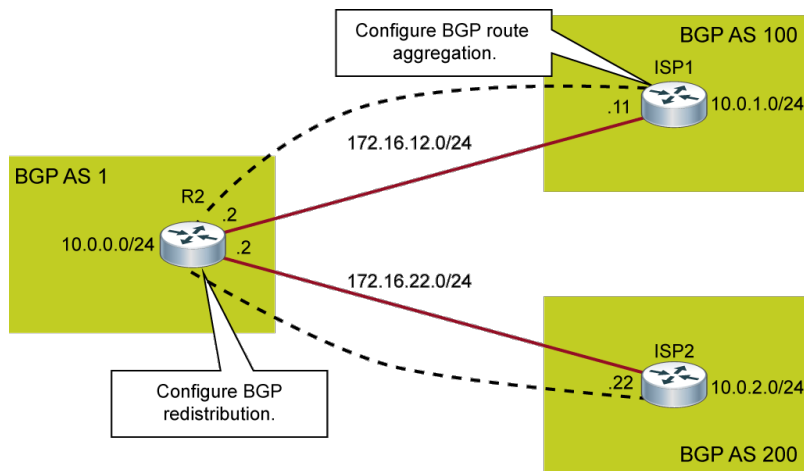
If there are a lot of networks to be advertised, and BGP is used primarily to achieve scalability, not routing security (for example, in enterprise networks), it could be easier to let the local [IGP](#) find the routes and then redistribute them into BGP. However, this approach introduces the risk that the IGP may find some networks that are not supposed to be advertised. Private network numbers, such as network 10.0.0.0/8, are often used within an [AS](#) for various reasons but must never be advertised out to the Internet. Careful filtering must be done to prevent unintentional advertising.

When the router injects a route that is listed with a **network** command into its BGP table, the origin code is set to "IGP." If the route is injected into the BGP table through redistribution, the origin code is set to "unknown/incomplete."

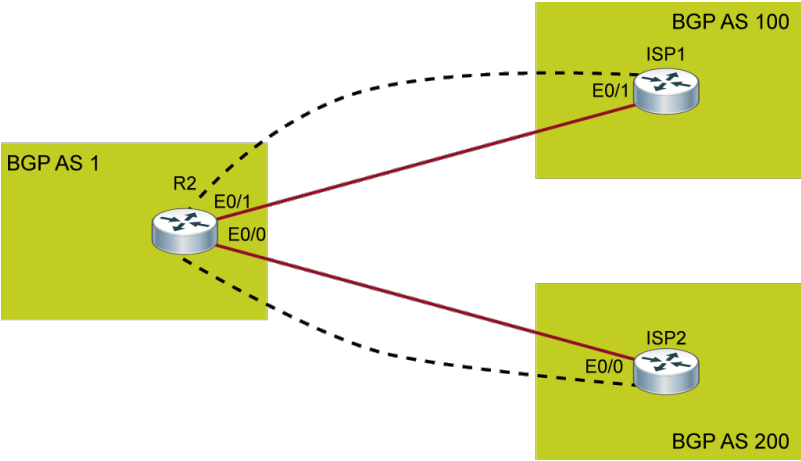
Discovery 2: Announcing Networks in BGP

Overview

Through this discovery, you will learn how to configure [BGP](#) redistribution and BGP route aggregation.



Topology



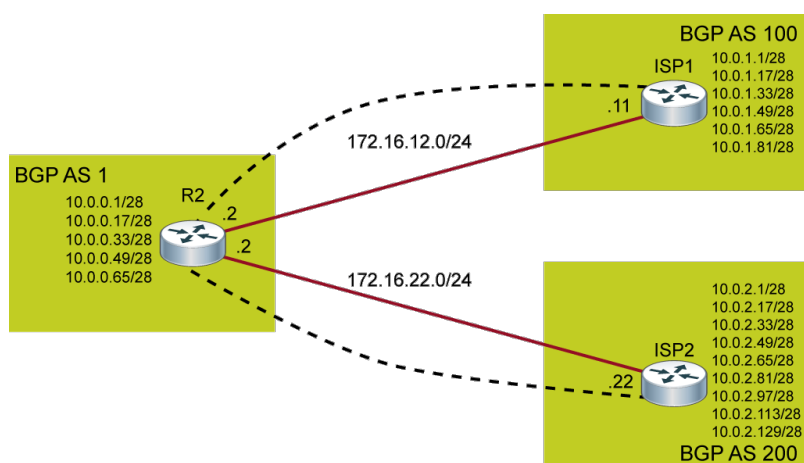
Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks



Announcing Networks in BGP

Discovery Steps

Step 1

You will redistribute connected routes into BGP. R2 router is preconfigured with five Loopback interfaces and is already running BGP in the [AS 1](#).

Initially, verify which Loopback interfaces are on the R2 router. Read Loopback interface's [IP addresses](#) and subnet masks. You will need this information when configuring access list.

```
R2# show ip interface | include Loopback|Internet address
< text omitted >
Loopback1 is up, line protocol is up
  Internet address is 10.0.0.1/28
Loopback2 is up, line protocol is up
  Internet address is 10.0.0.17/28
Loopback3 is up, line protocol is up
  Internet address is 10.0.0.33/28
Loopback4 is up, line protocol is up
  Internet address is 10.0.0.49/28
Loopback5 is up, line protocol is up
  Internet address is 10.0.0.65/28
```

On the R2 router, redistribute connected routes into BGP:

```
R2(config)# router bgp 1
R2(config-router)# redistribute connected
```

This way, you will include all connected interfaces, including Loopback interfaces, into the BGP routing table.

Step 2

On the R2 router, configure access list that permits even-numbered Loopback interfaces (Loopback 2 and Loopback 4).

Use access list number 1.

```
R2(config)# access-list 1 permit 10.0.0.16
R2(config)# access-list 1 permit 10.0.0.48
```

Step 3

On the R2 router, make sure that only even-numbered Loopback interfaces are redistributed into BGP.

```
R2(config)# router bgp 1
R2(config-router)# distribute-list 1 out connected
```

Step 4

Clear all BGP sessions on the R2 router.

```
R2# clear ip bgp *
```

Step 5

On the R2 router, verify that two routes are added into BGP routing table.

```
R2# show ip bgp
BGP table version is 18, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.16/28      0.0.0.0          0         32768 ?
```

*>	10.0.0.48/28	0.0.0.0	0	32768	?
*>	10.0.1.0/28	172.16.12.11	0	0	100 i
*>	10.0.1.16/28	172.16.12.11	0	0	100 i
*>	10.0.1.32/28	172.16.12.11	0	0	100 i
*>	10.0.1.48/28	172.16.12.11	0	0	100 i
*>	10.0.1.64/28	172.16.12.11	0	0	100 i
*>	10.0.1.80/28	172.16.12.11	0	0	100 i
*>	10.0.2.0/28	172.16.22.22	0	0	200 i
*>	10.0.2.16/28	172.16.22.22	0	0	200 i
*>	10.0.2.32/28	172.16.22.22	0	0	200 i
*>	10.0.2.48/28	172.16.22.22	0	0	200 i
*>	10.0.2.64/28	172.16.22.22	0	0	200 i
*>	10.0.2.80/28	172.16.22.22	0	0	200 i
*>	10.0.2.96/28	172.16.22.22	0	0	200 i
*>	10.0.2.112/28	172.16.22.22	0	0	200 i
*>	10.0.2.128/28	172.16.22.22	0	0	200 i

When you see next hop is 0.0.0.0, you know that these routes are locally inserted into BGP routing.

Also observe that origin code is incomplete for routes that are inserted by R2 router.

Redistribution Using Route Maps

Routes that are redistributed into BGP will carry the origin attribute incomplete. In most cases, this situation does not jeopardize BGP functionality. It could pose a problem if the route selection process has to decide on the best route toward a particular destination based on the [MED](#) attribute. In the case of receiving two routes, one with the [IGP](#) origin (inserted with the **network** command), and another one with the incomplete origin, the first route would always be selected, no matter what value the MED attribute is set to (according to the BGP route selection rules).

- Origin can be set to "IGP" with a route-map.
- Other BGP path attributes can also be set:
 - Metric
 - Next-hop
 - Community

You can configure route-maps on the router to filter updates and modify various attributes. A configured route-map can be applied to routes being redistributed from the IGP.

Only the routes that are permitted by the route-map will be redistributed. When you use the **set** command in the **route-map**, you can modify specific path attributes that are attached to the redistributed routes. Thus, only selected routes will be advertised, and they will have the desired attribute values.

When you configure the route-map, use a name. This name is a case-sensitive string, which is used when you are referring to the route-map. Any string could be used, but a meaningful name is suggested.

Use the **route-map** global configuration command and the **match** and **set** route-map configuration commands to define the conditions for redistributing routes. Each repetition of the **route-map** command has a list of **match** and **set** commands that are associated with it. The **match** commands specify the match criteria—the conditions under which redistribution is allowed for the current **route-map** command. The **set** commands specify the set actions—the particular redistribution actions to perform if the criteria enforced by the **match** commands are met.

When you are passing routes through a route-map, it can have several parts. Any route that does not match at least one match clause relating to a **route-map** command will be ignored; that is, the route will not be advertised. If you want to modify only some data, you must configure a second route-map section with an explicit match specified.

Step 6

On the R2 router, configure route map, where you will match based on previously configured access list and set origin code to IGP.

```
R2(config)# route-map evenLoop permit 10
R2(config-route-map)# match ip address 1
R2(config-route-map)# set origin igp
```

Step 7

On the R2 router, redistribute connected routes into BGP by using previously configured route map.

```
R2(config)# router bgp 1
R2(config-router)# redistribute connected route-map evenLoop
```

Step 8

On the R2 router, verify that origin code of two added routes in the BGP routing table has changed into IGP.

```
R2# show ip bgp | include 0.0.0.0
*> 10.0.0.16/28      0.0.0.0          0          32768 i
*> 10.0.0.48/28     0.0.0.0          0          32768 i
```

Configuring Classless BGP

BGP version 4 is a classless protocol, meaning that its routing updates include the IP address and the subnet mask. The combination of the IP address and the subnet mask is called an IP prefix. An IP prefix can be a subnet, a major network, or a supernet.

BGP uses prefix notation (address/number of bits) to display IP prefixes. The number following the slash (/) in the 192.168.0.0/16 notation in the figure refers to the number of bits in the subnet mask being set to 1. The subnet mask 255.255.0.0 starts with 16 consecutive bits set to 1, and the rest of the bits set to 0.

As another example, the subnet 172.16.1.0 with mask 255.255.255.0 can be written using the prefix notation as 172.16.1.0/24.

When classless prefix notation is used, an old class A network, for example, 10.0.0.0, with the natural mask, is written as 10.0.0.0/8. A class B network, 172.17.0.0 with natural mask, is written as 172.17.0.0/16, and a class C network, 192.168.1.0 with natural mask, is written as 192.168.1.0/24.

- BGP supports CIDR.
- Any BGP router can advertise individual networks or supernets (prefixes).
- Prefix notation is used with BGP instead of subnet masks.
 - 192.168.0.0/16 = 192.168.0.0 255.255.0.0

Step 9

On the R2 router, configure Loopback 1 interface subnet to be advertised into BGP. The Loopback 1 interface has a classless prefix preconfigured (10.0.0.0/28).

To advertise classless networks into BGP (a subnet or a supernet), you can use the **network** command with the **mask** keyword and the subnet mask specified. Suppose that an exact match is not found in the IP routing table. This situation can occur for example, when you are creating a summary or when you are advertising only a part of your address space. For the advertisement to succeed, you will need to manually configure a matching prefix on the router in the form of a static route pointing to the null 0 interface.

```
R2(config)# router bgp 1
R2(config-router)# network 10.0.0.0 mask 255.255.255.240
```

Step 10

On the R2 router, verify that Loopback 1 interface subnet is present in the BGP routing table.

```
R2# show ip bgp
BGP table version is 21, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	10.0.0.0/28	0.0.0.0	0		32768	i
*>	10.0.0.16/28	0.0.0.0	0		32768	i
*>	10.0.0.48/28	0.0.0.0	0		32768	i
*>	10.0.1.0/28	172.16.12.11	0		0 100	i
*>	10.0.1.16/28	172.16.12.11	0		0 100	i
*>	10.0.1.32/28	172.16.12.11	0		0 100	i
*>	10.0.1.48/28	172.16.12.11	0		0 100	i
*>	10.0.1.64/28	172.16.12.11	0		0 100	i
*>	10.0.1.80/28	172.16.12.11	0		0 100	i
*>	10.0.2.0/28	172.16.22.22	0		0 200	i
*>	10.0.2.16/28	172.16.22.22	0		0 200	i
*>	10.0.2.32/28	172.16.22.22	0		0 200	i
*>	10.0.2.48/28	172.16.22.22	0		0 200	i
*>	10.0.2.64/28	172.16.22.22	0		0 200	i
*>	10.0.2.80/28	172.16.22.22	0		0 200	i
*>	10.0.2.96/28	172.16.22.22	0		0 200	i
*>	10.0.2.112/28	172.16.22.22	0		0 200	i
*>	10.0.2.128/28	172.16.22.22	0		0 200	i

```
R2# show ip bgp 10.0.0.0/28
BGP routing table entry for 10.0.0.0/28, version 21
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    3
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.0.0.65)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
```

If the keyword **mask** and the subnet mask are omitted, the network is assumed to have its natural mask according to the network class. In this case, the route will still be injected into the BGP table on the router if there is any subnet of the major network that is reachable according to the routing table.

Step 11

On the R2 router, advertise subnet 10.0.0.0/24 into BGP. Subnet 10.0.0.0/24 is representing all Loopback interfaces that are configured on R2 router.

```
R2(config)# router bgp 1
R2(config-router)# network 10.0.0.0 mask 255.255.255.0
```

Step 12

On the R2 router, verify that subnet 10.0.0.0/24 is present in the BGP routing table.

```
R2# show ip bgp 10.0.0.0/24
% Network not in table
```

Route is not present in the BGP routing table.

The **network** command with the **mask** option tells BGP that 10.0.0.0/24 is a candidate for being advertised. The mask keyword and the mask 255.255.255.0 are required because the mask is not the natural one. However, before the candidate route is actually advertised, the router checks the routing table for an exact match (both network number and mask). It will always be found because there is a static route for it. This static route points to the null interface, which is always available.

The conclusion is that 10.0.0.0/24 will always be advertised by this router. All other BGP routers will use this information and forward any IP packets with the destination IP address in the interval 10.0.0.0 to 10.0.0.255 (inclusive) in the direction of this router. When those packets arrive, the router, in this example, must have more explicit routes to the different parts of the 10.0.0.0/24 address range. IGP could answer this need but it is not shown in this configuration example.

Step 13

On the R2 router, configure explicit static route for 10.0.0.0/24 pointing to the null interface.

```
R2(config)# ip route 10.0.0.0 255.255.255.0 Null 0
```

If, however, an IP packet arrives with a destination address to which this router does not have a more explicit route, the static route will route the packet to the null interface, where it is dropped. This routing is a safety precaution that will prevent a routing loop, which might occur when route summaries are used in combination with default routing. Suppose that a packet arrives from the Internet to a subnet of 10.0.0.0/24, which is currently not reachable. In this case, the packet might follow the default route toward the Internet because there is no more explicit route. Of course, the packet would immediately be routed back again, and a routing loop would occur.

Step 14

On the R2 router, verify that subnet 10.0.0.0/24 is now present in the BGP routing table.

```
R2# show ip bgp 10.0.0.0/24
BGP routing table entry for 10.0.0.0/24, version 22
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    3
  Refresh Epoch 1
  Local
    0.0.0.0 from 0.0.0.0 (10.0.0.65)
      Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local, best
```

Route 10.0.0.0/24 is present in the BGP routing table.

Aggregating BGP Networks

When the BGP table is already populated with routes that should be summarized, you must configure a router to do so. The summarization of BGP routes is called "aggregation."

Use aggregation when a group of more specific routes has been injected into the BGP table at one stage but can be summarized at a later stage. The routes to be summarized could be IGP routes that have been redistributed into BGP. Before BGP advertises these routes to the rest of the network, an aggregation of the subnets into a larger announcement would be appropriate.

In some networks, more specific routes are injected into the BGP table by some routers, and aggregation is done in another router or even in another AS. This approach is called "proxy aggregation."

When a router is configured to do aggregation, you must configure the route summary. If any route that is already in the BGP table is within the range that is indicated by the summary, then the summary route is also injected into the BGP table on the router and advertised to other routers. This action creates more information in the BGP table. To get any benefits from the aggregation, you must suppress the more specific routes that are covered by the route summary. This suppression is an option to the **aggregate** configuration command.

When you suppress the more specific routes through configuration, they are still present in the BGP table of the router doing the aggregation. However, because the routes are marked as suppressed, they are never advertised to any other router.

Summarization is called "aggregation" in BGP.

- Aggregation creates summary routes (called "aggregates") from networks already in BGP table.
- Individual networks can be announced or suppressed.

Step 15

On the ISP1 router, configure BGP aggregation on the subnet 10.0.1.0/24.

Do not use **summary-only** keyword, to observe both the route summary, and the more specific routes that will be advertised.

```
ISP1(config)# router bgp 100
ISP1(config-router)# aggregate-address 10.0.1.0 255.255.255.0
```

Step 16

On the ISP1 router, verify presence of aggregated route 10.0.1.0/24 in the BGP routing table.

```
ISP1# show ip bgp
BGP table version is 73, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28       172.16.12.2          0             0 1 i
*> 10.0.0.0/24       172.16.12.2          0             0 1 i
*> 10.0.0.16/28      172.16.12.2          0             0 1 i
*> 10.0.0.48/28      172.16.12.2          0             0 1 i
*> 10.0.1.0/28       0.0.0.0              0            32768 i
*> 10.0.1.0/24       0.0.0.0              0            32768 i
*> 10.0.1.16/28      0.0.0.0              0            32768 i
*> 10.0.1.32/28      0.0.0.0              0            32768 i
*> 10.0.1.48/28      0.0.0.0              0            32768 i
*> 10.0.1.64/28      0.0.0.0              0            32768 i
*> 10.0.1.80/28      0.0.0.0              0            32768 i
*> 10.0.2.0/28       172.16.12.2          0             0 1 200 i
*> 10.0.2.16/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.32/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.48/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.64/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.80/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.96/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.112/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.128/28     172.16.12.2          0             0 1 200 i
```

Step 17

On the R2 router, verify presence of aggregated route 10.0.1.0/24 and more specific routes in the BGP routing table.

```
R2# show ip bgp
BGP table version is 37, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28       0.0.0.0              0            32768 i
*> 10.0.0.0/24       0.0.0.0              0            32768 i
*> 10.0.0.16/28      0.0.0.0              0            32768 i
*> 10.0.0.48/28      0.0.0.0              0            32768 i
*> 10.0.1.0/28       172.16.12.11          0             0 100 i
*> 10.0.1.0/24       172.16.12.11          0             0 100 i
*> 10.0.1.16/28      172.16.12.11          0             0 100 i
*> 10.0.1.32/28      172.16.12.11          0             0 100 i
*> 10.0.1.48/28      172.16.12.11          0             0 100 i
*> 10.0.1.64/28      172.16.12.11          0             0 100 i
*> 10.0.1.80/28      172.16.12.11          0             0 100 i
*> 10.0.2.0/28       172.16.22.22          0             0 200 i
*> 10.0.2.16/28      172.16.22.22          0             0 200 i
*> 10.0.2.32/28      172.16.22.22          0             0 200 i
*> 10.0.2.48/28      172.16.22.22          0             0 200 i
*> 10.0.2.64/28      172.16.22.22          0             0 200 i
*> 10.0.2.80/28      172.16.22.22          0             0 200 i
*> 10.0.2.96/28      172.16.22.22          0             0 200 i
*> 10.0.2.112/28     172.16.22.22          0             0 200 i
*> 10.0.2.128/28     172.16.22.22          0             0 200 i
```

This approach is generally not desirable. Therefore, suppression of individual routes, described next, is used in most cases.

Step 18

On the ISP1 router, configure BGP aggregation on the subnet 10.0.1.0/24 by using keyword **summary-only**.

```
ISP1(config)# router bgp 100
ISP1(config-router)# aggregate-address 10.0.1.0 255.255.255.0 summary-only
```

Step 19

On the ISP1 router, verify presence of aggregated route 10.0.1.0/24 in the BGP routing table.

```
ISP1# show ip bgp
BGP table version is 79, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28      172.16.12.2          0             0 1 i
*> 10.0.0.0/24      172.16.12.2          0             0 1 i
*> 10.0.0.16/28     172.16.12.2          0             0 1 i
*> 10.0.0.48/28     172.16.12.2          0             0 1 i
s> 10.0.1.0/28      0.0.0.0              0          32768 i
*> 10.0.1.0/24      0.0.0.0              0          32768 i
s> 10.0.1.16/28     0.0.0.0              0          32768 i
s> 10.0.1.32/28     0.0.0.0              0          32768 i
s> 10.0.1.48/28     0.0.0.0              0          32768 i
s> 10.0.1.64/28     0.0.0.0              0          32768 i
s> 10.0.1.80/28     0.0.0.0              0          32768 i
*> 10.0.2.0/28      172.16.12.2          0             0 1 200 i
*> 10.0.2.16/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.32/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.48/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.64/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.80/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.96/28     172.16.12.2          0             0 1 200 i
*> 10.0.2.112/28    172.16.12.2          0             0 1 200 i
*> 10.0.2.128/28    172.16.12.2          0             0 1 200 i
```

The prefix 10.0.1.0/24 is injected because there is at least one more specific route within the summary range. In this case 10.0.1.0/28, 10.0.1.16/28, 10.0.1.32/28, 10.0.1.48/28, 10.0.1.64/28, and 10.0.1.80/28 are within the range. The more specific routes are marked as suppressed using the lowercase letter "s." The "s" means that they are still present and available in the BGP table of the router, but they are not advertised on any BGP session.

When the summary-only option is used, only the route summary will be advertised, not the more specific routes.

One of the benefits of this approach is that the rest of the routers will receive only one route instead of many more specific routes. It eases the burden on the other routers by reducing the amount of memory that is required to hold the BGP table.

Another benefit is that route flapping is reduced. The router doing the aggregation continues advertising the aggregate as long as there is at least one specific route within the range still available. If one of the more specific routes is lost but at least one remains, the aggregate itself is not lost. The flap of the more specific route is not visible to the rest of the network. This approach reduces the number of updates necessary and the CPU power that is required to process them.

However, all route summarization in any routing protocol causes a loss of granularity (that is, lack of more detailed network or subnet visibility). Suboptimal routing could be introduced when redundant paths are available to reach a group of networks that are advertised by a single route summary. Some of the networks could be more reachable via one of the paths, while others may be more reachable another way. From outside the immediate network, multiple paths may not be visible because only summary routes are advertised. Therefore, there is a risk that the least optimal path will be chosen.

Step 20

On the R2 router, verify that only BGP aggregated route is received from ISP1 router.

You will not see more specific routes (10.0.1.0/28, 10.0.1.16/28, 10.0.1.32/28, 10.0.1.48/28, 10.0.1.64/28, or 10.0.1.80/28) advertised from ISP1 router.

```
R2# show ip bgp
BGP table version is 29, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.0.0/28      0.0.0.0             0         32768 i
*> 10.0.0.0/24      0.0.0.0             0         32768 i
*> 10.0.0.16/28     0.0.0.0             0         32768 i
*> 10.0.0.48/28     0.0.0.0             0         32768 i
*> 10.0.1.0/24      172.16.12.11        0           0 100 i
*> 10.0.2.0/28      172.16.22.22        0           0 200 i
*> 10.0.2.16/28     172.16.22.22        0           0 200 i
*> 10.0.2.32/28     172.16.22.22        0           0 200 i
*> 10.0.2.48/28     172.16.22.22        0           0 200 i
*> 10.0.2.64/28     172.16.22.22        0           0 200 i
*> 10.0.2.80/28     172.16.22.22        0           0 200 i
*> 10.0.2.96/28     172.16.22.22        0           0 200 i
*> 10.0.2.112/28    172.16.22.22        0           0 200 i
*> 10.0.2.128/28    172.16.22.22        0           0 200 i
```


BGP Conditional Route Injection

Routes that are advertised through the [BGP](#) are commonly aggregated to minimize the number of routes that are used and reduce the size of global routing tables. However, common route aggregation can obscure more specific routing information that is more accurate but not necessary to forward packets to their destinations. Common route aggregation obscures routing accuracy because a prefix that represents multiple addresses or hosts over a large topological area cannot be accurately reflected in a single route. Cisco IOS Software provides several methods by which you can originate a prefix into BGP. The methods include redistribution and using the **network** or **aggregate-address** command. These methods assume the existence of more specific routing information (matching the route to be originated) in either the routing table or the BGP table.

The BGP Conditional Route Injection feature has the following functions:

- It provides means to originate a prefix into a BGP routing table without the corresponding match.
- It allows more specific routes to be generated based on administrative policy or traffic engineering information to provide more specific control over the forwarding of packets to these more specific routes, which are injected into the BGP routing table only if the configured conditions are met.
- It improves accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes.

The BGP Conditional Route Injection feature allows you to originate a prefix into a BGP routing table without the corresponding match. This feature allows more specific routes to be generated based on administrative policy or traffic engineering information. This feature provides more specific control over the forwarding of packets to these more specific routes. These routes are injected into the BGP routing table only if the configured conditions are met. Enabling this feature allows you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix may be injected.

The BGP Conditional Route Injection feature is enabled with the **bgp inject-map exist-map** command. This command uses two route maps (inject-map and exist-map) to install one (or more than one) more specific prefix into a BGP routing table. The exist-map specifies the prefixes that the BGP speaker will track. The inject-map defines the prefixes that will be created and installed into the local BGP table.

BGP Conditional Route Injection Example

This configuration example configures conditional route injection for the inject-map that is named ORIGINATE and the exist-map that is named LEARNED_PATH:

```
router bgp 109
  bgp inject-map ORIGINATE exist-map LEARNED_PATH
  !
  route-map LEARNED_PATH permit 10
    match ip address prefix-list ROUTE
    match ip route-source prefix-list ROUTE_SOURCE
  !
  route-map ORIGINATE permit 10
    set ip address prefix-list ORIGINATED_ROUTES
    set community 14616:555 additive
  !
  ip prefix-list ROUTE permit 10.1.1.0/24
  !
  ip prefix-list ORIGINATED_ROUTES permit 10.1.1.0/25
  ip prefix-list ORIGINATED_ROUTES permit 10.1.1.128/25
  !
  ip prefix-list ROUTE_SOURCE permit 10.2.1.1/32
```

BGP Support for TTL Security Check

The BGP Support for TTL Security Check feature introduces a lightweight security mechanism to protect EBGp peering sessions from CPU utilization-based attacks. These types of attacks are typically brute-force DoS attacks that attempt to disable the network by flooding the network with IP packets that contain forged source and destination IP addresses. This feature protects the EBGp peering session by comparing the value in the TTL field of received IP packets against a hop count. The hop count is configured locally for each EBGp peering session. If the value in the TTL field of the incoming IP packet is greater than or equal to the locally configured value, the IP packet is accepted and processed normally. If the TTL value in the IP packet is less than the locally configured value, the packet is silently discarded and no ICMP message is generated. Not responding to a forged packet is part of a designed behavior that helps prevent CPU utilization-based attacks.

BGP Support for TTL Security Check feature has the following characteristics:

- Lightweight security mechanism to protect EBGp peering sessions from CPU utilization-based attacks.
- Protects the EBGp peering session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each EBGp peering session.
- Supports both directly connected peering sessions and multihop EBGp peering sessions.

Accurately forging the TTL count in an IP packet is generally considered to be impossible. It is possible to forge the TTL field in an IP packet header. However, accurately forging a packet to match the TTL count from a trusted peer is not possible unless the network to which the trusted peer belongs has been compromised.

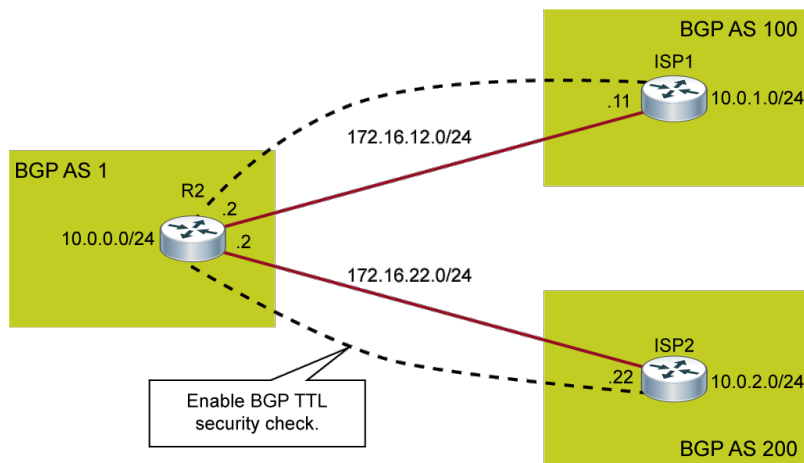
This feature supports both directly connected peering sessions and multihop EBGp peering sessions. The incoming packets that contain invalid TTL values do not affect the BGP peering session. The BGP peering session remains open, and the router silently discards the invalid packet. The BGP session, however, can still expire if keepalive packets are not received before the session timer expires.

The BGP Support for TTL Security Check feature should be configured on each participating router. It provides an effective and easy-to-deploy solution to protect EBGp peering sessions from CPU utilization-based attacks. When this feature is enabled, a host cannot attack a BGP session if the host is not a member of the local or remote BGP network or if the host is not directly connected to a network segment between the local and remote BGP networks. This solution greatly reduces the effectiveness of DoS attacks against a BGP AS.

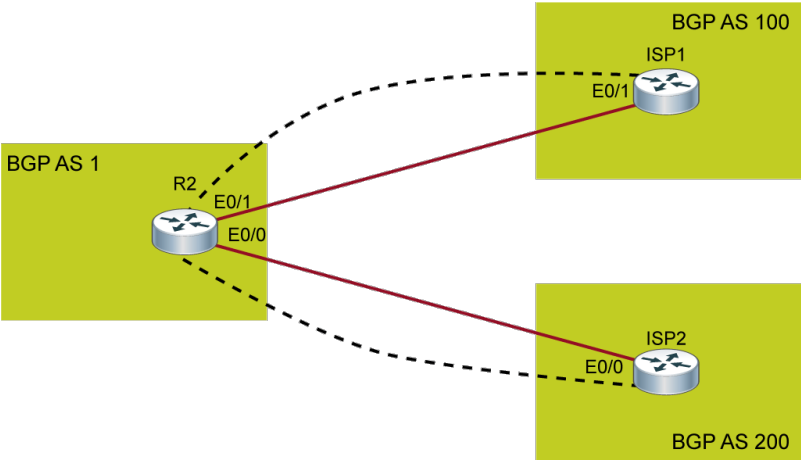
Discovery 3: Implement BGP TTL Security Check

Overview

Through this discovery, you will learn how to enable [BGP TTL](#) security check.



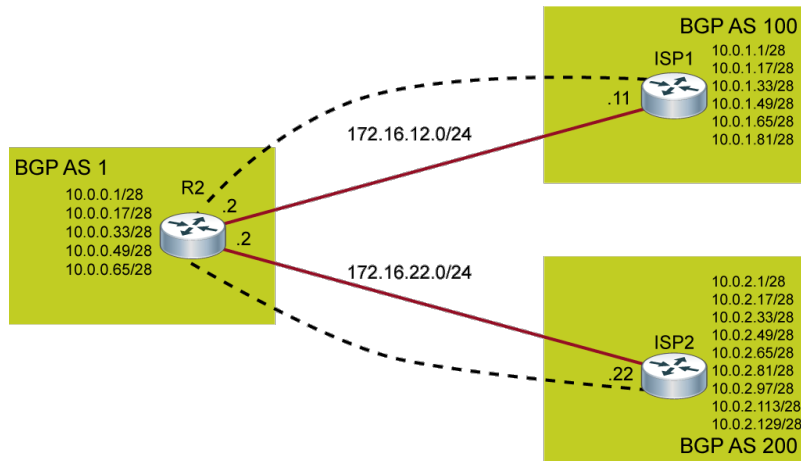
Topology



Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."



Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

Implement BGP TTL Security Check

Discovery Steps

Step 1

To secure a BGP peering session between R2 and ISP2 routers configure BGP TTL security check.

On the R2 router, configure BGP TTL security check to the ISP2 with hop count 10.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.22.2 ttl-security hops 10
```

BGP session between R2 and ISP2 routers will go down.

Router R2 will start sending BGP packets with TTL 255 and will accept BGP packets with 245 or more.

Step 2

On the ISP2 router, configure BGP TTL security check to the R2 with hop count 10.

```
ISP2(config)# router bgp 200
ISP2(config-router)# neighbor 172.16.22.2 ttl-security hops 10
```

BGP session between R2 and ISP2 routers will go up.

```
%BGP-5-ADJCHANGE: neighbor 172.16.22.2 Up
```

Step 3

On the ISP2 router, verify what TTL values are used for BGP neighbor.

```
ISP2# show ip bgp neighbors 172.16.22.2 | include TTL
Connection is ECN Disabled, Minimum incoming TTL 245, Outgoing TTL 255
```

You can see BGP packets are sent with TTL number 255 and are accepted with TTL value 245 or more.

Step 4

On the R2 router, verify what TTL values are used for BGP neighbor.

```
R2# show ip bgp neighbors 172.16.22.22 | include TTL
Connection is ECN Disabled, Minimum incoming TTL 245, Outgoing TTL 255
```

You can see BGP packets are sent with TTL number 255 and are accepted with TTL value 245 or more.

Step 5

On the R2 router, verify that BGP session to the ISP2 is established.

```
R2# show ip bgp summary
BGP router identifier 10.0.0.65, local AS number 1
BGP table version is 63, main routing table version 63
14 network entries using 2072 bytes of memory
14 path entries using 896 bytes of memory
3/3 BGP path/bestpath attribute entries using 408 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3424 total bytes of memory
BGP activity 38/24 prefixes, 38/24 paths, scan interval 60 secs

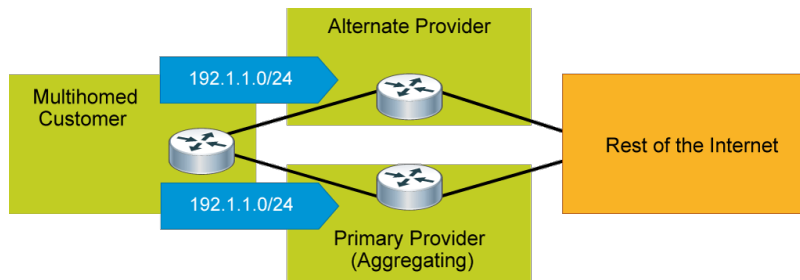
Neighbor      V      AS MsgRcvd MsgSent  TblVer  InQ  OutQ  Up/Down  State/PfxRcd
```

172.16.12.11	4	100	241	247	63	0	0	00:37:51	1
172.16.22.22	4	200	19	23	63	0	0	00:12:47	9

BGP session between R2 and ISP2 routers is secured with BGP TTL security check.

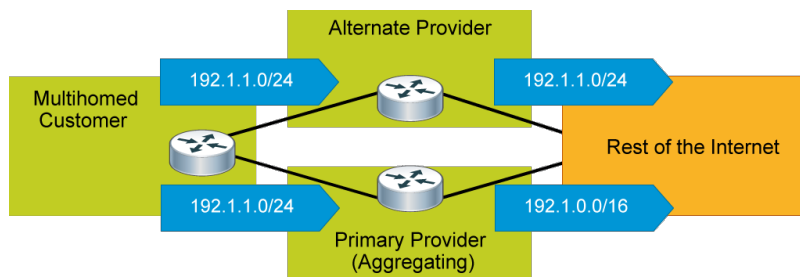
Multihomed Customer Problem

In this example, the primary provider is doing aggregation of 192.1.0.0/16 before sending it to the rest of the network. This situation means that the primary provider is also doing proxy aggregation for the route 192.1.1.0/24 that is advertised by the multihomed customer. The rest of the Internet will not see the route 192.1.1.0/24 via the primary provider.



- Customer prefers primary provider, using alternate only as backup.
- Primary provider advertises the aggregate.
- Alternate provider advertises individual network.

The multihomed customer also advertises 192.1.1.0/24 to the alternate provider. In this case, the provider does not do any aggregation of any routes starting with 192.1 (and should not do so). This situation means that the alternate provider will propagate 192.1.1.0/24 to the rest of the Internet.



- Customer prefers primary provider, using alternate only as backup.
- Primary provider advertises the aggregate.
- Alternate provider advertises individual network.

The rest of the Internet now sees overlapping routes. It sees 192.1.1.0/24 as reachable via the alternate provider and 192.1.0.0/16 as reachable via the primary provider. These two routes are treated as different routes. They are not compared with each other in a route selection process because they indicate different destinations. Because the router views them as different destinations, both routes will be injected into the routing table.

If a packet arrives with a destination address in the 192.1.1.0/24 network, the rest of the Internet will follow the "longest matching prefix" rule and forward the packet to the alternate provider.

To avoid this issue, the primary provider must turn off aggregation. If the primary provider does so, the rest of the Internet will see 192.1.1.0/24 both ways. And, because the same route (network and mask) is reachable in two ways, route selection processing starts. Depending on the attribute values, the rest of the Internet could be advised to use the primary provider instead of the alternate one.

However, turning off aggregation will also cause the primary provider to advertise all routes within the aggregate, and all benefits of aggregation will be lost.

Summary

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

- The BGP process in a Cisco router is started with the **router bgp** command and BGP neighbors are added with **neighbor** command.
- MD5 authentication can be used to secure a connection between two BGP neighbors. BGP also supports TTL security check feature, which protects EBGP sessions from CPU utilization-based attacks.
- Local networks are announced in BGP by listing them with the **network** command or by redistributing them.
- BGP supports aggregation but BGP route aggregation is not appropriate in multihomed topologies.

Monitoring and Troubleshooting BGP

Overview

To ensure that basic [BGP](#) configurations are operating correctly, you need to be familiar with BGP monitoring commands. If basic BGP configurations are not functioning as expected, BGP troubleshooting skills are critical to successful problem resolution.

This lesson introduces the Cisco IOS commands that are available for monitoring and troubleshooting basic BGP configurations. The commands that are required to monitor the status of BGP, neighbor connections, and the BGP table are discussed. The lesson also discusses techniques for troubleshooting the most common BGP session startup issues.

Upon completing this lesson, you will be able to:

- Identify the Cisco IOS command that is required to monitor the overall status of the BGP routing process
- Identify the Cisco IOS command that is required to monitor BGP neighbors
- Identify the Cisco IOS commands that are required to monitor the BGP table
- Identify the Cisco IOS commands that are required to perform basic BGP debugging
- List common BGP session startup problems
- Troubleshoot basic BGP session startup problems when the neighbor is not reachable
- Troubleshoot basic BGP session startup problems when the neighbor is not configured
- Troubleshoot basic BGP session startup problems when an AS number mismatch exists

Monitoring Overall BGP Routing

To display the status of all [BGP](#) connections, use the **show ip bgp summary** EXEC command.

This command has no arguments or keywords.

```
router> show ip bgp summary
```

- Displays BGP memory use, and displays BGP neighbors and the state of communication with them

```
R2# show ip bgp summary
BGP router identifier 172.16.22.2, local AS number 1
BGP table version is 10, main routing table version 10
9 network entries using 1332 bytes of memory
9 path entries using 576 bytes of memory
1/1 BGP path/bestpath attribute entries using 136 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2068 total bytes of memory
BGP activity 9/0 prefixes, 9/0 paths, scan interval 60 secs

Neighbor      V    AS MsgRcvd MsgSent   TblVer  InQ  OutQ Up/Down  State/P
fxRcd
172.16.12.11   4   100      0      0       1    0    0 never   Idle
172.16.22.22   4   200     63     62      10    0    0 00:54:24
9
```

This command is very useful when you are troubleshooting BGP. The output in the figure provides a short summary of the status of the BGP process in the router.

The first section of the output describes the BGP table and its content:

- The BGP table version is the version number of the local BGP table. This number is increased every time that the table is changed.
- The main routing table version shows the last version of the BGP database that was injected into the main routing table.
- The subsequent lines of text indicate the amount of memory that has been allocated to hold the table. These lines of text display how many networks are known and how many different paths and attribute values are associated with them.

The second section of the output is a table in which the current neighbor statuses are shown. There is one line of text for each neighbor that has been configured. The columns are as follows:

- IP address of the neighbor as configured in the local router
- BGP version number that the router uses when communicating with the neighbor
- [AS](#) number of the remote neighbor
- Number of messages and updates that have been received from the neighbor since the session was established
- Number of messages and updates that have been sent to the neighbor since the session was established
- Version number of the local BGP table that has been included in the most recent update to the neighbor
- Number of messages that are waiting to be processed in the incoming queue from this neighbor
- Number of messages that are waiting in the outgoing queue for transmission to the neighbor
- How long the neighbor has been in the current state and the name of the current state (the state "Established" is not printed out, so no state name indicates "Established")

You can use this information to verify that BGP sessions are up and established. If they are not, you will have to further investigate the BGP configuration to locate the problem. You can also verify the IP address and AS number of the configured BGP neighbor with the **show ip bgp summary** command.

If the session state is "Established," the number of messages that have been sent

and received, as displayed in the output of the **show ip bgp summary** command, can indicate BGP stability. Use the command a few times, with a time interval between the printouts, and calculate how many messages have been exchanged during that period.

Many messages in the incoming queue indicate a lack of CPU resources in the local router. Many messages in the outgoing queue indicate a lack of bandwidth to the remote router or a lack of CPU resources in the remote router.

Monitoring BGP Neighbors

To display information about the [TCP](#) and [BGP](#) connections to neighbors, use the **show ip bgp neighbors EXEC** command.

```
router> show ip bgp neighbors ip-address
```

- Displays detailed neighbor information

```
R2# show ip bgp neighbors 172.16.12.11
BGP neighbor is 172.16.12.11, remote AS 100, external link
  BGP version 4, remote router ID 10.0.1.81
  BGP state = Established, up for 00:03:10
  Last read 00:00:26, last write 00:00:31, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
```

You can use this command for two different purposes. The general purpose, as shown in the figure, is to get information about the TCP session and the BGP parameters of the session. All BGP session parameters are displayed. In addition, TCP timers and counters are also displayed.

The other use is not shown in this example. If any of the optional qualifiers referring to routes or paths are given, the BGP routing information that was sent or received on this session is displayed. This feature is useful when you are troubleshooting path selection.

Monitoring BGP Table

To display entries in the [BGP](#) routing table, use the **show ip bgp EXEC** command.

```
router> show ip bgp
```

- Displays all routes in the BGP table in summary format

```
R2# show ip bgp
BGP table version is 18, local router ID is 172.16.22.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  1.0.0.0/24      172.16.22.22             0           0 200 i
*    172.16.12.11             0           0 100 99 i
*>  10.0.1.0/28     172.16.12.11             0           0 100 i
*>  10.0.1.16/28    172.16.12.11             0           0 100 i
*>  10.0.1.32/28    172.16.12.11             0           0 100 i
*>  10.0.1.48/28    172.16.12.11             0           0 100 i
*>  10.0.1.64/28    172.16.12.11             0           0 100 i
*>  10.0.1.80/28    172.16.12.11             0           0 100 i
*>  10.0.2.0/28     172.16.22.22             0           0 200 i
<... output omitted ...>
```

In most cases, when the **show ip bgp** command is given without optional qualifiers, the entire BGP table is displayed. An abbreviated list of information about each route is displayed, one line per prefix. The output is sorted in network number order. Therefore, if the BGP table contains more than one route to the same network, the routes are displayed on successive lines. The network number is printed on the first of these lines only. The following lines, which refer to the same network, have the network number field left blank.

Some, but not all, of the BGP attributes that are associated with the route are displayed on the line. Next-hop, [MED](#) (displayed as "Metric"), local preference, and weight each have their own columns. The AS-path attribute is displayed as the sequence of [AS](#) numbers in the "Path" column. Immediately following the AS path, but not part of the AS-path attribute, the origin attribute is displayed. The lowercase letter "i" means [IGP](#), "e" means [EGP](#), and "?" means incomplete or unknown.

The BGP path selection process selects one of the available routes to each of the networks as the best. These routes are pointed out by the character ">" in the left column.

```
router> show ip bgp ip-prefix [mask subnet-mask]
```

- Displays detailed information about all paths for a single prefix

```
R2# show ip bgp 1.0.0.1
BGP routing table entry for 1.0.0.0/24, version 18
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  200
    172.16.22.22 from 172.16.22.22 (10.0.2.129)
      Origin IGP, metric 0, localpref 100, valid, external, best
  Refresh Epoch 1
  100 99
    172.16.12.11 from 172.16.12.11 (10.0.1.81)
      Origin IGP, metric 0, localpref 100, valid, external
```

- BGP attributes are listed for each route: AS-Path, Next-Hop, advertising router IP address, advertising router Router-ID.
- The bottom row lists other BGP attributes.

If more information and the complete set of BGP attributes are required, the **show ip bgp** command should be entered with the network number on the command line. This command displays all relevant BGP information about that specific network.

In this example, the information about network 1.0.0.1 is displayed. There are two routes to 1.0.0.1. One is received from neighbor 172.16.12.11 and the other from 172.16.22.22.

The BGP route selection process has selected this route as the best. So, BGP will try to install this route into the routing table. Installation of routes in the routing table is made based on the [AD](#).

Debugging BGP

There are several Cisco IOS commands that you can use to perform debugging of basic [BGP](#) configurations.

```
router# debug ip tcp transactions
```

- Displays all TCP transactions (start of session, session errors, etc.)

```
router# debug ip bgp events
```

- Displays significant BGP events (neighbor state transitions, update runs)

If a BGP session stays in the Active state, where it is actively sending connection attempts to the neighbor, **debug ip tcp transactions** can provide valuable information about failed connection attempts. All [TCP](#) transactions in the router are displayed on the console as they happen. The network administrator can then determine whether the TCP session is being established, and, if not, what the probable cause of the problem might be.

If the TCP session succeeds but is torn down within a short time, you might find the reason if you use **debug ip bgp events**. All BGP events will be displayed on the console as they happen if this debug command is enabled.

```
router# debug ip bgp keepalives
```

- Debugs BGP keepalive packets

```
router# debug ip bgp updates
```

- Displays all incoming or outgoing BGP updates
- *Use with caution*

In a stable state with no network topology changes, no BGP updates are sent between neighboring routers. When a BGP session has been idle for some time, the BGP protocol will exchange keepalive packets between BGP neighbors. The keepalive timer has a default value of 60 seconds.

Use the **debug ip bgp keepalives** command to get a printout on the console for every keepalive packet that is sent or received. Successful keepalive exchanges indicate that the session is working and is in a stable state.

If no keepalives have been sent or received, the session might still be working. The reason for not seeing any keepalives would be that the session is never idle long enough.

Use the **debug ip bgp updates** command to get a printout on the console for every update message that is sent or received. The successful exchange of updates indicates that the session is working and is not in the Idle state.

In a large network, updates are sent and received in large volumes. Starting the **debug ip bgp updates** command might cause extensive output on the console. In some cases, the CPU resources that are used to generate those outputs are so great that few CPU resources remain to actually forward traffic. In a case with very busy BGP sessions, it is actually possible to set the router in a condition where all CPU resources are consumed with the debugging printouts.

```
router# debug ip bgp updates acl
```

- Displays all incoming or outgoing BGP updates for routes matching an IP access-list

```
router# debug ip bgp ip-address updates [acl]
```

- Displays all BGP updates that are received from or sent to a BGP neighbor (optionally matching an IP access-list)

To avoid debug printouts for every update that is sent or received, you can create and associate an access-list with the **debug** command. When you use this command, the console displays only the updates that refer to a network number that

is permitted by the access-list. The command is extremely useful in a live network with busy BGP sessions where the troubleshooter is interested only in updates for specific networks.

Indicating a specific neighbor can even further restrict the debugging. The console displays only the updates on the session with the indicated neighbor. Optionally, you can combine this debug command with an access-list.

BGP Session Startup Problems

There are several common session startup issues that you can experience when configuring basic [BGP](#).

Common BGP session startup symptoms:

- BGP neighbors do not become active.
- BGP neighbor is active, but the session is never established.
- BGP neighbor oscillates between idle and active.

BGP Neighbor Not Reachable

BGP sessions to a router in another AS should normally run across directly connected interfaces (routers that share a common IP subnet). You must configure neighboring routers to reach each other using the IP address belonging to this shared subnet so that no other routing protocol is required to set up the BGP session.

If a router is configured with a BGP neighbor that is in another AS but not directly connected, the session will stay in the Idle state. The router will not even attempt to set up the session.

Symptom:

- BGP neighbors do not become active.
 - **show ip bgp neighbors** displays the neighbor state as Idle for several minutes.

Diagnosis:

- Neighbor is not directly connected.

Verification:

- Verify with **show ip route**.

The normal way to fix this problem is to change the neighbor reference so that it is referred by an IP address that is directly connected. However, in some odd cases, the neighbor is intentionally reachable using an interface that is not directly connected. In that rare case, the local router must have routing information on how to reach that address. Also, you must configure the BGP session with the **ebgp-multihop** option.

If the session goes into the Active state, the router will attempt to establish the session. If session establishment is unsuccessful, you will have to troubleshoot the problem. The **debug ip tcp transactions** command will display the connect attempts.

Symptom:

- BGP neighbor is active; session is not established.
 - **debug ip tcp transactions** display shows that the TCP SYN packet is not answered with a SYN-ACK packet.

Diagnosis:

- Neighbor is not reachable.

Verification:

- Verify connectivity with **ping**.
- Check for the presence of an access-list.

TCP session establishment starts with the router sending a TCP SYN packet. If the TCP SYN packet is never answered, the remote router might be dead or not reachable. Try to use the **ping** command and verify the existence of the remote router and the IP packet exchange between the local and remote router.

BGP Neighbor Not Reachable Example

In this example, the remote BGP router is not available.

```
ISP1# debug ip tcp transactions
TCP special event debugging is on
<... output omitted ...>
*Feb  6 05:04:01: TCBEFE6AF18 created
*Feb  6 05:04:01: TCBEFE6AF18 setting property TCP_VRFTABLEID (20) EC3C8C6
C
<... output omitted ...>
*Feb  6 05:04:01: TCP: Random local port generated 43645, network 1
*Feb  6 05:04:01: TCBEFE6AF18 bound to 172.16.12.11.43645
*Feb  6 05:04:01: Reserved port 43645 in Transport Port Agent for TCP IP t
ype 1
*Feb  6 05:04:01: TCP: sending SYN, seq 2161553223, ack 0
*Feb  6 05:04:01: TCP0: Connection to 172.16.12.22:179, advertising MSS 14
60
*Feb  6 05:04:01: TCP0: state was CLOSED -> SYNSENT [43645 -
> 172.16.12.22(179)]
<... ouput omitted ...>
```

```
*Feb  6 05:04:31: TCP0: state was SYNSENT -> CLOSED [43645 -  
> 172.16.12.22(179)]  
*Feb  6 05:04:31: TCB 0xEFE6AF18 destroyed
```

- SYN packet is sent.
- SYN-ACK reply never comes back. TCP session is closed.

The sending router never receives the reply to the SYN packet and aborts the TCP session in approximately 45 seconds (changing the state from synsent to closed).

BGP Neighbor Not Configured

If the [TCP SYN](#) packet is answered with a TCP [RST](#) packet, the remote router is alive and reachable but is not willing to grant the connection attempt. The reason for this refusal may be that [BGP](#) has not been fully configured on the remote router or that the source IP address that is used by the local router in the connection attempt is not in the list of valid neighbors for the remote router.

Symptom:

- BGP neighbor is active; session is not established.
 - **debug ip tcp transactions** display shows that the TCP SYN packet is answered with an RST packet.

Diagnosis:

- This router is not configured as the BGP neighbor on the neighboring router.

Verification:

- Check IP addresses of BGP neighbors with **show ip bgp summary** on the neighboring router.

BGP Neighbor Not Configured Example

In this example, the remote router is not configured for BGP or there was a mismatch in the neighbor IP addresses.

```
ISP2# debug ip tcp transactions
*Feb  9 00:35:12.675: TCBEFA28EF0 created
<... output omitted ...>
*Feb  9 00:35:12: TCBEFA28EF0 bound to 172.16.22.2.45207
*Feb  9 00:35:12: Reserved port 45207 in Transport Port Agent for TCP IP t
ype 1
*Feb  9 00:35:12: TCP: sending SYN, seq 4270019552, ack 0
*Feb  9 00:35:12: TCP0: Connection to 172.16.22.2:179, advertising MSS 146
0
*Feb  9 00:35:12: TCP0: state was CLOSED -> SYNSENT [45207 -
> 172.16.22.2(179)]
*Feb  9 00:35:12: Released port 45207 in Transport Port Agent for TCP IP t
ype 1 delay 240000
*Feb  9 00:35:12: TCP0: state was SYNSENT -> CLOSED [45207 -
> 172.16.22.2(179)]
*Feb  9 00:35:12: TCP0: bad seg from 172.16.22.2 -
- closing connection: port 45207 seq 0 ack 4270019553 rcvnxt 0 rcvwnd 0 le
n 0
*Feb  9 00:35:12: TCP0: connection closed - remote sent RST
*Feb  9 00:35:12: TCB 0xEFA28EF0 destroyed
```

- SYN packet is sent.
- Neighbor replies with an RST packet. TCP session is closed.

The remote router responds with an RST packet as soon as it receives the initial SYN packet, terminating the BGP session.

BGP AS Number Mismatch

If the [TCP](#) session is established using the specified three-way handshake of [SYN](#), [SYN-ACK](#), and [ACK](#), but the router drops the session after a short packet exchange, the [BGP](#) parameters are mismatched. Make sure that the remote [AS](#) that is configured on the router matches the local AS that is configured on the neighbor. If the AS numbers do not match, the router drops the session after exchanging BGP Open messages.

Symptom:

- BGP neighbor oscillates between Active and Idle.
 - **debug ip tcp transactions** displays the TCP session being established and torn down immediately.

Diagnosis:

- There is an AS number mismatch between BGP neighbors.

Verification:

- Verify the AS numbers that are configured for neighboring routers using the **show ip bgp summary** on both routers.

BGP AS Number Mismatch Example

This example illustrates a mismatch in an AS number.

```
R2# debug ip tcp transaction
R2# debug ip bgp event

*Feb  6 06:37:44: TCBEE9C5A38 created
*Feb  6 06:37:44: TCP0: state was LISTEN -> SYNRCVD [179 -
> 172.16.12.11(60291)]
*Feb  6 06:37:44: TCP: tcb EE9C5A38 connection to 172.16.12.11:60291, peer
MSS 1460, MSS is 516
*Feb  6 06:37:44: TCP: sending SYN, seq 2355663235, ack 879693158
*Feb  6 06:37:44: TCP0: Connection to 172.16.12.11:60291, advertising MSS
1460
*Feb  6 06:37:44: TCP0: state was SYNRCVD -> ESTAB [179 -
> 172.16.12.11(60291)]
*Feb  6 06:37:44: TCBEE9C4B20 accepting EE9C5A38 from 172.16.12.11.60291
*<... output omitted ...>
*Feb  6 06:37:44: %BGP-3-
NOTIFICATION: sent to neighbor 172.16.12.11 passive 2/2 (peer in wrong AS)
2 bytes 0064
*<... output omitted ...>
*Feb  6 06:37:44: TCP0: FIN processed
*Feb  6 06:37:44: TCP0: state was ESTAB -> CLOSEWAIT [179 -
> 172.16.12.11(60291)]
*<... output omitted ...>
*Feb  6 06:37:49: TCP0: sending FIN
```

- First, TCP session is established.
- BGP notification is sent because of AS number mismatch.

Whenever there is a mismatch in AS numbers (or any other BGP parameters that are necessary for proper BGP operation), the BGP session is terminated with a BGP notification. The TCP session is terminated as well.

Summary

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

- The **show ip bgp summary** command displays the overall status of BGP and shows configured neighbors and their state.
- You can use the **show ip bgp neighbors** command to get more in-depth information about a specific BGP neighbor.
- All entries in the BGP table can be displayed with the **show ip bgp** command. You can also use **show ip bgp** to display an extended printout about a specific route in the BGP table.
- You can use the **debug ip tcp transactions** command to troubleshoot BGP session establishment problems. The command **debug ip bgp events** displays significant BGP events, while **debug ip bgp updates** displays the routing information being exchanged between BGP neighbors.

Module Summary

Overview

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

- BGP has reliable transport provided by TCP, a rich set of metrics called BGP path attributes, and scalability features such as batched updates that make it suitable for very large networks.
- Configured BGP neighbors establish a TCP session and exchange the BGP Open message, which contains the parameters that each BGP router proposes to use.
- Some path attributes are well-known and should be recognized by every BGP implementation. Some of the well-known attributes, such as AS-path, next-hop, and origin, are mandatory and have to be present in every BGP update.
- The route selection process takes into account various BGP attributes that are attached to the route, as well as local decisions.
- When you are configuring BGP neighbors, you will enable the BGP routing protocol process, establish neighbors, and advertise local routes.

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Border Gateway Protocol*.
http://docwiki.cisco.com/wiki/Border_Gateway_Protocol
- Cisco Systems, Inc. *Configuring BGP*.
http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfbgp.html
- Cisco Systems, Inc. *Using the Border Gateway Protocol for Interdomain Routing*
http://docwiki.cisco.com/wiki/Internetworking_Case_Studies

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 the AS is true? (Source: "Introducing BGP")

☐ The AS is a collection of networks under a single administrative domain.

☐ The AS is a collection of networks that belong to different enterprise networks.

☐ The AS requires IGP protocol to exchange routing information between autonomous systems.

☐ The AS number can be any number between 0 and 255.

1. Which routing method best describes BGP? (Source: "Introducing BGP")

- ☐ distance vector
- ☐ link state
- ☐ path vector
- ☐ hybrid of link state and distance vector

1. A transit AS is an AS that exchanges BGP routing information with other autonomous systems and forwards information that is received from one AS to another AS. True or false? (Source: "Introducing BGP")

☐ true

☐ false

1. Which three items are BGP enhancements to traditional distance vector routing protocols? (Choose three.) (Source: "Introducing BGP")

☐ reliable updates

☐ use of triggered updates only

☐ enhanced security

☐ rich metrics

☐ route summarization

☐ snapshot updates

1. Which protocol facilitates reliable update capabilities in BGP?
(Source: "Introducing BGP")

☐ TCP

☐ UDP

☐ HSRP

☐ ICMP

1. What are three characteristics of an AS? (Choose three.) (Source: "Introducing BGP")

- ☐ AS uses IGPs for intradomain routing.
- ☐ AS uses EGPs for interdomain routing.
- ☐ AS is a collection of networks under a common administrative authority.
- ☐ AS consists of a group of network domains.
- ☐ AS automatically summarizes addresses
- ☐ AS is regulated by the IETF.

1. Which three scenarios are common scenarios where BGP is used? (Choose three.) (Source: "Introducing BGP")

- ☐ a customer with a connection to multiple service providers
- ☐ service provider networks acting as transit systems and forwarding external traffic through their network
- ☐ a single-site customer intranet with complex administrative policies between departments
- ☐ as the core routing protocol in very large enterprise networks
- ☐ as the routing protocol in an IS-IS backbone area
- ☐ as the core routing protocol in an SNA network

1. What are three recommended BGP use guidelines for multihomed customer networks? (Choose three.) (Source: "Introducing BGP")

☐ Most multihomed customers should use BGP with their service providers.

☐ Most multihomed customers should forward routing information that is received from one provider to the other provider.

☐ The multihomed customer must have its own public AS number.

☐ Multihomed customers should use a provider-independent, public address space.

☐ The multihomed customer may use and advertise RFC 1918 addresses.

☐ Multihomed customers should use the AS number of their primary ISP.

1. What is a limitation of the BGP routing protocol? (Source: "Introducing BGP")

- You cannot use BGP to implement hop-by-hop routing policy controls.

- You cannot use BGP to influence the routing policy in a downstream AS.

- BGP cannot control forwarding of packets based on their destination address.

- BGP cannot scale to very large networks with more than 110,000 routes.

1. Which three statements are true of BGP mandatory well-known attributes? (Choose three.) (Source: "Understanding BGP Path Attributes")

- ☐ They must be present in all BGP updates.
- ☐ All BGP-compliant implementations must recognize them.
- ☐ All BGP-compliant routers must adhere to policies specified in mandatory attributes.
- ☐ All well-known attributes are propagated to other neighbors.
- ☐ They must be present in some BGP updates.

1. Which three attributes are BGP mandatory well-known attributes? (Choose three.) (Source: "Understanding BGP Path Attributes")

☐ next-hop

☐ weight

☐ AS-path

☐ origin

☐ MED

☐ local preference

1. Which three possible values are assigned to the BGP origin attribute? (Choose three.) (Source: "Understanding BGP Path Attributes")

☐ IGP

☐ EGP

☐ unknown

☐ internal

☐ external

☐ MED

1. Which nontransitive optional BGP attribute is useful in assisting with the route selection process when multiple links to another AS exist? (Source: "Understanding BGP Path Attributes")

☐ next-hop

☐ local preference

☒ MED

☐ AS-path

1. In which two ways can the BGP next-hop attribute be modified? (Choose two.) (Source: "Understanding BGP Path Attributes")

☐ If the next-hop attribute is in the same IP subnet as the receiving router, the attribute is unchanged; otherwise, it is set to the IP address of the sending router.

☐ The next-hop attribute is always set to the IP address of the sending router.

☐ The next-hop attribute is modified only when BGP packets exit an AS.

☐ The BGP next-hop attribute is modified only when BGP packets traverse point-to-point links.

1. Which three statements regarding the BGP AS-path attribute are true? (Choose three.) (Source: "Understanding BGP Path Attributes")

☐ The local AS number is prepended to the AS path each time that the route crosses an AS boundary.

☐ The AS that originally injected the route into BGP is always found at the rightmost end of the AS path.

☐ The AS-path attribute can be used to avoid routing loops.

☐ BGP routes with an empty AS path were injected into BGP from outside the local AS.

☐ The local AS number is appended to the end of the AS path each time that the route crosses an AS boundary.

☐ The AS that originally injected the route into BGP is always found at the leftmost end of the AS path.

1. Which description applies to the local preference attribute? (Source: "Understanding BGP Path Attributes")

☐	well-known mandatory
☐	well-known discretionary
☐	optional transitive
☐	optional nontransitive

1. Which two attributes are BGP optional transitive attributes? (Choose two.) (Source: "Understanding BGP Path Attributes")

☐	MED
☐	weight
☐	aggregator
☐	atomic aggregate
☐	community

1. What is indicated by a state of "Idle" in the output of the **show ip bgp summary** command? (Source: "Establishing BGP Sessions")

• The router is currently not attempting to establish a connection with a neighbor.

• The connection to the configured neighbor has timed out.

• The connection to a BGP neighbor has been established, and no errors have been received on the connection.

• The connection to a BGP neighbor has been established, and no packets have been sent.

1. How are neighbors detected in BGP? (Source: "Establishing BGP Sessions")

• They are discovered automatically

• They must be configured manually.

1. Router-ID is a number that uniquely identifies the router. How is it selected? (Source: "Establishing BGP Sessions")

- Router-ID is the highest IP address of any loopback interface. If there is no loopback interface, the router-ID must be configured manually.

- Router-ID must be configured manually.

- Router-ID is the highest IP address of any loopback interface. If there is no loopback interface, the router uses the highest IP address of any interface.

- Router-ID is the lowest IP address of any loopback interface. If there is no loopback interface, the router uses the lowest IP address of any interface.

1. What happens if two TCP connection attempts between configured BGP neighbors succeed? (Source: "Establishing BGP Sessions")

Both connections will be terminated, and the neighbors will re-establish a neighbor relationship.

One connection will be maintained as primary and the other as backup.

One of the two connections will be torn down

The router with the lower router-ID will determine if the second connection is torn down or used as a backup TCP connection.

1. Given the following BGP session states, in which state is the router while waiting for the other router to respond to the BGP Open message? (Source: "Establishing BGP Sessions")

☐ OpenConfirm

☐ Established

☐ Idle

☐ OpenSent

☐ Active

1. Given the following BGP session states, what is their order of progression during the creation of a successful neighbor session? (Source: "Establishing BGP Sessions")

OpenConfirm	3
OpenSent	5
Established	4
Active	2
Idle	1

1. BGP sessions are established over TCP using port number 179.
True or false? (Source: "Establishing BGP Sessions")

☒ true

☐ false

1. What does the field "TblVer" indicate in the output of the **show ip bgp summary** command? (Source: "Establishing BGP Sessions")

- ☐ the current version of BGP in use by the router
- ☐ the number of route prefixes that are contained in the BGP update of the router
- ☐ BGP messages that have been received from that neighbor
- ☐ the last version of the BGP database that was sent to that neighbor

1. What occurs when you use MD5 between two BGP neighbors?
(Source: "Establishing BGP Sessions")

- Every packet is encrypted with MD5.

- The IP header is encrypted using MD5.

- An MD5 checksum is calculated and sent with each packet so that its source can be verified.

- A username and password are embedded in an IP datagram that is matched to a username and password on the remote neighbor.

1. What does a router that is running BGP do with a BGP update that contains its own AS path? (Source: "Processing BGP Routes")

☐ The router checks to see whether the information that is contained in the update is better than its current information. If it is, it will update its BGP table.

☐ The router accepts the route update.

☐ The router silently discards (denies) the route.

☐ The router returns an error to the router that sent the update.

1. Which three attributes must always be present in the routing update? (Choose three.) (Source: "Processing BGP Routes")

☐	MED
☐	AS-path
☐	local-preference
☐	next-hop
☐	origin

1. When a router has more than one alternative route to reach the same destination, local preference is checked before the weight attribute. True or false? (Source: "Processing BGP Routes")

☒ true

☐ false

1. How many alternate paths to a single destination will a BGP router maintain in the BGP table? (Source: "Processing BGP Routes")

The router will maintain only the best path to the destination.

The router will maintain two paths, the best path and a backup route.

The BGP table will hold up to four routes by default and a maximum of six configurable routes.

The BGP table will store all valid, advertised routes to the destination in the BGP table.

1. When a router has more than one alternative route to reach the same IP subnet (network and mask), the router has to select one of them as best in its default mode of operation Match the following steps to the correct step in the process. (Source: "Processing BGP Routes")

The router compares MED values, but only if it receives the updates from the same neighboring AS. Routes with a lower MED are preferred.

Step 1

The lengths of the AS paths are compared (the content is not checked; only the number of autonomous systems in each AS path is counted). The route with the shortest length is selected.

Step 5

If the AS-path lengths are the same, the origin code is checked. BGP prefers the path with the lowest origin type: IGP is lower than EGP, and EGP is lower than Incomplete.

Step 4

If the local preference attributes are different, the route with the highest value is selected as best.

Step 2

The router prefers the route with the higher weight.

Step 6

The router checks whether the next-hop attribute indicates an IP address that is reachable according to the current routing table. If the next hop is not reachable, the router does not consider the BGP route as a candidate to become selected as the best.

Step 3

If one of the routes is injected into the BGP table by the local router, the local router prefers it to any routes that it receives from other BGP routers.

Step 7

1. What are two ways in which local networks are advertised into the BGP routing protocol process? (Choose two.) (Source: "Processing BGP Routes")

- ☐ automatically, after a BGP neighbor session is established
- ☐ manually, with the **network** command
- ☐ through redistribution into the BGP process
- ☐ by advertising them to the BGP table on the router after Cisco Discovery Protocol discovers connected networks

1. What are two situations when is it appropriate to disable automatic summarization in BGP? (Choose two.) (Source: "Processing BGP Routes")

- ☐ when BGP neighbors are not configured to advertise aggregate routes to upstream providers
- ☐ when the classless variant of the **network** command is used
- ☐ when you are using a classless IGP in the AS
- ☐ when the effects of automatic summarization of IGP-to-BGP redistribution are not desired

1. What is the AD of BGP routes in the IP routing table that were learned from BGP neighbors in a different AS? (Source: "Processing BGP Routes")

1

20

90

120

1. Which three BGP attributes are displayed for each route in the BGP table when you are using the **show ip bgp** command? (Choose three.) (Source: "Processing BGP Routes")

☐ weight

☐ communities

☐ origin

☐ AS-path

1. What is the valid AS number range for a BGP process on a Cisco router? (Source: "Configuring Basic BGP")

☐ 1 to 256

☐ 1 to 32768

☐ 1 to 65535

☐ 1 to 131072

1. Which two parameters must you configure with the **neighbor** command to establish a BGP session with an external neighbor? (Choose two.) (Source: "Configuring Basic BGP")

☐ neighbor IP address

☐ subnet mask of the IP network

☐ remote AS number

☐ local AS number

☐ description of the neighbor

1. What is the best method to temporarily disable a BGP neighbor session? (Source: "Configuring Basic BGP")

Remove the **neighbor** command from the BGP router process.

Remove the BGP router process from the configuration.

Terminate the neighbor connection with the **neighbor shutdown** command

Disconnect the neighbor by initiating a router reload.

1. Which two of the following statements about configuring BGP timers are accurate? (Choose two.) (Source: "Configuring Basic BGP")

☐ Changing the BGP default holdtime and keepalive timers is usually not recommended.

☐ The **neighbor timers** command sets the timers for a specific BGP peer or peer group.

☐ The **timers bgp** command sets the timers for a specific BGP peer or peer group.

☐ Holdtime indicates the frequency (in seconds) with which the Cisco IOS software sends messages to its peer.

1. Which two of the following are characteristics of the string component of the **neighbor** {*ip-address* | *peer-group-name*} **password** *string* command? (Choose two.) (Source: "Configuring Basic BGP")

- ☐ can contain any alphanumeric characters, including spaces
- ☐ case-sensitive password of up to 100 characters
- ☐ first character can be a number
- ☐ cannot specify a password in the format "number-space-anything"

1. Which three steps must you complete to advertise a classless prefix into BGP? (Choose three.) (Source: "Configuring Basic BGP")

- ☐ Configure the prefix with the **network** command.
- ☐ Specify the **mask** keyword with the locally advertised route.
- ☐ Configure the **redistribute connected** command under the BGP router process.
- ☐ Use a static route pointing to null 0 that matches the prefix.

1. Which origin code is carried with routes that are redistributed into BGP? (Source: "Configuring Basic BGP")

☐ internal

☐ external

☐ unknown

☐ incomplete

1. Which two of the following statements about the classless behavior of BGP are correct? (Choose two.) (Source: "Configuring Basic BGP")

☐ When an exact match is not found in the IP routing table a matching prefix is automatically configured on the router.

☐ In the **network** *ip-prefix-address* **mask** *subnet-mask* command, the prefix does not have to match an entry in the IP routing table.

☐ To advertise classless networks into BGP (a subnet or a supernet), you can use the **network** command with the **mask** keyword and the subnet mask specified.

☐ If the keyword **mask** and the subnet mask are omitted, the network is assumed to have its natural mask according to the network class.

1. What are two benefits of using route aggregation in BGP? (Choose two.) (Source: "Configuring Basic BGP")

☐ It ensures that even if aggregate networks are down, the aggregate is advertised, which eliminates black holes.

☐ It reduces the amount of memory that is used in the router to store the BGP table.

☐ It reduces route flapping and its effects on router CPU resources.

☐ BGP attribute granularity is maintained, which ensures optimal path selection.

1. Which two of the following are characteristics of the BGP Conditional Route Injection feature? (Choose two.) (Source: "Configuring Basic BGP")

- ☐ allows you to originate a prefix into a BGP routing table without the corresponding match
- ☐ enabled with the **bgp inject-map exist-map** command
- ☐ allows conditional injecting or replacing more specific prefixes with less specific prefixes
- ☐ allows origination of a prefix into a BGP routing table only with the corresponding match

1. Which two of the following are characteristics of the BGP Support for TTL Security Check feature? (Choose two.) (Source: "Configuring Basic BGP")

- ☐ should be configured on only one participating router
- ☐ prevents BGP sessions from expiring even if keepalive packets are not received before the session timer expires
- ☐ protects the EBGP peering session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each EBGP peering session
- ☐ supports both directly connected peering sessions and multihop EBGP peering sessions

1. Which two of the following are functions of the **show ip bgp summary** command? (Choose two.)

- ☐ displays BGP memory use
- ☐ displays BGP neighbors and status of communication with them
- ☐ locates problems in BGP sessions that are up and established
- ☐ displays the BGP routing table

1. Which command do you use to display detailed BGP neighbor information? (Source: "Monitoring and Troubleshooting BGP")

☐ **show ip bgp summary**

☐ **show ip bgp**

☐ **show ip bgp neighbors address**

☐ **show ip bgp detail**

1. To get information about the TCP session and BGP parameters of the session, you can issue **show ip bgp summary** command. True or false? (Source: "Monitoring and Troubleshooting BGP")

☒ true

☐ false

1. Which two of the following statements about the **show ip bgp** command that is used to monitor the BGP routing table are accurate? (Choose two.) (Source: "Monitoring and Troubleshooting BGP")

- ☐ The **show ip bgp** command shows an abbreviated list of information about each route, displaying one line per prefix.
- ☐ The **show ip bgp** command shows a full list of information about each route, displaying one line per prefix.
- ☐ All of the BGP attributes that are associated with the route are displayed on the line.
- ☐ If the BGP table contains more than one route to the same network, the routes are displayed on successive lines of the command output.

1. Which debug command should you enable to troubleshoot BGP session startup issues where the TCP connection never succeeds? (Source: "Monitoring and Troubleshooting BGP")

☐ **ip bgp updates**

☐ **ip packets**

☐ **ip bgp keepalives**

☐ **ip tcp transactions**

1. When debugging an exchange of updates, you should use **debug ip bgp keepalive** command. True or False?

☒ true

☐ false

1. What are the three most common session startup issues that you can experience when configuring basic BGP? (Choose three.)
(Source: "Monitoring and Troubleshooting BGP")

- ☐ BGP neighbors do not become active.
- ☐ BGP routing loops cause black holes.
- ☐ A BGP neighbor is active, but the BGP session is not established.
- ☐ The BGP neighbor state oscillates between Idle and Active.
- ☐ The BGP session is active, but the neighbor cannot be reached.
- ☐ BGP keepalives experience intermittent failures.

1. What is the most common reason for a BGP session not leaving the Idle state? (Source: "Monitoring and Troubleshooting BGP")

- The TCP port for the connection is not configured.
- The external neighbor is not directly connected.
- The TCP SYN packet is answered with an RST packet.
- The neighbors have been configured with the same AS number.

1. What will result from attempting to open a BGP connection with a neighbor that has not been properly configured for BGP? (Source: "Monitoring and Troubleshooting BGP")

• The BGP session will remain in the Idle state.

• The neighbor session will be established, and the session startup parameters will be negotiated over the TCP session.

• The BGP session will be immediately terminated with a TCP RST packet.

• The BGP session will become "stuck in Active state."

1. When a BGP neighbor oscillates between Active and Idle, what is the likely diagnosis? (Source: "Monitoring and Troubleshooting BGP")

There are mismatched keepalive intervals.

There is an AS number mismatch between BGP neighbors.

One router is not configured as the BGP neighbor on the neighboring router.

The BGP neighbor is not reachable.

Answer Key

1. Which statement about the AS is true? (Source: "Introducing BGP")



The AS is a collection of networks under a single administrative domain.



The AS is a collection of networks that belong to different enterprise networks.



The AS requires IGP protocol to exchange routing information between autonomous systems.



The AS number can be any number between 0 and 255.

1. Which routing method best describes BGP? (Source: "Introducing BGP")

- ☐ distance vector
- ☐ link state
- ☐ path vector
- ☐ hybrid of link state and distance vector

1. A transit AS is an AS that exchanges BGP routing information with other autonomous systems and forwards information that is received from one AS to another AS. True or false? (Source: "Introducing BGP")

☒ true

☐ false

1. Which three items are BGP enhancements to traditional distance vector routing protocols? (Choose three.) (Source: "Introducing BGP")

☒	reliable updates
☒	use of triggered updates only
☐	enhanced security
☒	rich metrics
☐	route summarization
☐	snapshot updates

1. Which protocol facilitates reliable update capabilities in BGP?
(Source: "Introducing BGP")

☐ TCP

☐ UDP

☐ HSRP

☐ ICMP

1. What are three characteristics of an AS? (Choose three.) (Source: "Introducing BGP")

- ☒ AS uses IGPs for intradomain routing.
- ☒ AS uses EGP for interdomain routing.
- ☒ AS is a collection of networks under a common administrative authority.
- ☐ AS consists of a group of network domains.
- ☐ AS automatically summarizes addresses
- ☐ AS is regulated by the IETF.

1. Which three scenarios are common scenarios where BGP is used? (Choose three.) (Source: "Introducing BGP")

- ☒ a customer with a connection to multiple service providers
- ☒ service provider networks acting as transit systems and forwarding external traffic through their network
- ☐ a single-site customer intranet with complex administrative policies between departments
- ☒ as the core routing protocol in very large enterprise networks
- ☐ as the routing protocol in an IS-IS backbone area
- ☐ as the core routing protocol in an SNA network

1. What are three recommended BGP use guidelines for multihomed customer networks? (Choose three.) (Source: "Introducing BGP")



Most multihomed customers should use BGP with their service providers.



Most multihomed customers should forward routing information that is received from one provider to the other provider.



The multihomed customer must have its own public AS number.



Multihomed customers should use a provider-independent, public address space.



The multihomed customer may use and advertise RFC 1918 addresses.



Multihomed customers should use the AS number of their primary ISP.

1. What is a limitation of the BGP routing protocol? (Source: "Introducing BGP")

- You cannot use BGP to implement hop-by-hop routing policy controls.

- You cannot use BGP to influence the routing policy in a downstream AS.

- BGP cannot control forwarding of packets based on their destination address.

- BGP cannot scale to very large networks with more than 110,000 routes.

1. Which three statements are true of BGP mandatory well-known attributes? (Choose three.) (Source: "Understanding BGP Path Attributes")

- ☒ They must be present in all BGP updates.
- ☒ All BGP-compliant implementations must recognize them.
- ☐ All BGP-compliant routers must adhere to policies specified in mandatory attributes.
- ☒ All well-known attributes are propagated to other neighbors.
- ☐ They must be present in some BGP updates.

1. Which three attributes are BGP mandatory well-known attributes? (Choose three.) (Source: "Understanding BGP Path Attributes")

☒ next-hop

☐ weight

☒ AS-path

☒ origin

☐ MED

☐ local preference

1. Which three possible values are assigned to the BGP origin attribute? (Choose three.) (Source: "Understanding BGP Path Attributes")

☒ IGP

☒ EGP

☒ unknown

☐ internal

☐ external

☐ MED

1. Which nontransitive optional BGP attribute is useful in assisting with the route selection process when multiple links to another AS exist? (Source: "Understanding BGP Path Attributes")

☐ next-hop

☐ local preference

☒ MED

☐ AS-path

1. In which two ways can the BGP next-hop attribute be modified? (Choose two.) (Source: "Understanding BGP Path Attributes")



If the next-hop attribute is in the same IP subnet as the receiving router, the attribute is unchanged; otherwise, it is set to the IP address of the sending router.



The next-hop attribute is always set to the IP address of the sending router.



The next-hop attribute is modified only when BGP packets exit an AS.



The BGP next-hop attribute is modified only when BGP packets traverse point-to-point links.

1. Which three statements regarding the BGP AS-path attribute are true? (Choose three.) (Source: "Understanding BGP Path Attributes")

☒ The local AS number is prepended to the AS path each time that the route crosses an AS boundary.

☒ The AS that originally injected the route into BGP is always found at the rightmost end of the AS path.

☒ The AS-path attribute can be used to avoid routing loops.

☐ BGP routes with an empty AS path were injected into BGP from outside the local AS.

☐ The local AS number is appended to the end of the AS path each time that the route crosses an AS boundary.

☐ The AS that originally injected the route into BGP is always found at the leftmost end of the AS path.

1. Which description applies to the local preference attribute? (Source: "Understanding BGP Path Attributes")

☐	well-known mandatory
☐	well-known discretionary
☐	optional transitive
☐	optional nontransitive

1. Which two attributes are BGP optional transitive attributes? (Choose two.) (Source: "Understanding BGP Path Attributes")

☐	MED
☐	weight
☒	aggregator
☐	atomic aggregate
☒	community

1. What is indicated by a state of "Idle" in the output of the **show ip bgp summary** command? (Source: "Establishing BGP Sessions")

The router is currently not attempting to establish a connection with a neighbor.

The connection to the configured neighbor has timed out.

The connection to a BGP neighbor has been established, and no errors have been received on the connection.

The connection to a BGP neighbor has been established, and no packets have been sent.

1. How are neighbors detected in BGP? (Source: "Establishing BGP Sessions")

☐ They are discovered automatically

☐ They must be configured manually.

1. Router-ID is a number that uniquely identifies the router. How is it selected? (Source: "Establishing BGP Sessions")

Router-ID is the highest IP address of any loopback interface. If there is no loopback interface, the router-ID must be configured manually.

Router-ID must be configured manually.

Router-ID is the highest IP address of any loopback interface. If there is no loopback interface, the router uses the highest IP address of any interface.

Router-ID is the lowest IP address of any loopback interface. If there is no loopback interface, the router uses the lowest IP address of any interface.

1. What happens if two TCP connection attempts between configured BGP neighbors succeed? (Source: "Establishing BGP Sessions")

Both connections will be terminated, and the neighbors will re-establish a neighbor relationship.

One connection will be maintained as primary and the other as backup.

One of the two connections will be torn down

The router with the lower router-ID will determine if the second connection is torn down or used as a backup TCP connection.

1. Given the following BGP session states, in which state is the router while waiting for the other router to respond to the BGP Open message? (Source: "Establishing BGP Sessions")

OpenConfirm

Established

Idle

OpenSent

Active

1. Given the following BGP session states, what is their order of progression during the creation of a successful neighbor session? (Source: "Establishing BGP Sessions")

Idle	1
Active	2
OpenSent	3
OpenConfirm	4
Established	5

1. BGP sessions are established over TCP using port number 179. True or false? (Source: "Establishing BGP Sessions")

☒ true

☐ false

1. What does the field "TblVer" indicate in the output of the **show ip bgp summary** command? (Source: "Establishing BGP Sessions")

☐ the current version of BGP in use by the router

☐ the number of route prefixes that are contained in the BGP update of the router

☐ BGP messages that have been received from that neighbor

☐ the last version of the BGP database that was sent to that neighbor

1. What occurs when you use MD5 between two BGP neighbors?
(Source: "Establishing BGP Sessions")

- Every packet is encrypted with MD5.

- The IP header is encrypted using MD5.

- An MD5 checksum is calculated and sent with each packet so that its source can be verified.

- A username and password are embedded in an IP datagram that is matched to a username and password on the remote neighbor.

1. What does a router that is running BGP do with a BGP update that contains its own AS path? (Source: "Processing BGP Routes")

☐ The router checks to see whether the information that is contained in the update is better than its current information. If it is, it will update its BGP table.

☐ The router accepts the route update.

☐ The router silently discards (denies) the route.

☐ The router returns an error to the router that sent the update.

1. Which three attributes must always be present in the routing update? (Choose three.) (Source: "Processing BGP Routes")

☐	MED
☒	AS-path
☐	local-preference
☒	next-hop
☒	origin

1. When a router has more than one alternative route to reach the same destination, local preference is checked before the weight attribute. True or false? (Source: "Processing BGP Routes")

☐ true

☐ false

1. How many alternate paths to a single destination will a BGP router maintain in the BGP table? (Source: "Processing BGP Routes")

The router will maintain only the best path to the destination.

The router will maintain two paths, the best path and a backup route.

The BGP table will hold up to four routes by default and a maximum of six configurable routes.

The BGP table will store all valid, advertised routes to the destination in the BGP table.

1. When a router has more than one alternative route to reach the same IP subnet (network and mask), the router has to select one of them as best in its default mode of operation Match the following steps to the correct step in the process. (Source: "Processing BGP Routes")

The router compares MED values, but only if it receives the updates from the same neighboring AS. Routes with a lower MED are preferred.

Step 7

The router checks whether the next-hop attribute indicates an IP address that is reachable according to the current routing table. If the next hop is not reachable, the router does not consider the BGP route as a candidate to become selected as the best.

Step 1

The router prefers the route with the higher weight.

Step 2

If the local preference attributes are different, the route with the highest value is selected as best.

Step 3

The lengths of the AS paths are compared (the content is not checked; only the number of autonomous systems in each AS path is counted). The route with the shortest length is selected.

Step 5

If one of the routes is injected into the BGP table by the local router, the local router prefers it to any routes that it receives from other BGP routers.

Step 4

If the AS-path lengths are the same, the origin code is checked. BGP prefers the path with the lowest origin type: IGP is lower than EGP, and EGP is lower than Incomplete.

Step 6

1. What are two ways in which local networks are advertised into the BGP routing protocol process? (Choose two.) (Source: "Processing BGP Routes")

- ☐ automatically, after a BGP neighbor session is established
- ☒ manually, with the **network** command
- ☒ through redistribution into the BGP process
- ☐ by advertising them to the BGP table on the router after Cisco Discovery Protocol discovers connected networks

1. What are two situations when is it appropriate to disable automatic summarization in BGP? (Choose two.) (Source: "Processing BGP Routes")

☐ when BGP neighbors are not configured to advertise aggregate routes to upstream providers

☒ when the classless variant of the **network** command is used

☐ when you are using a classless IGP in the AS

☒ when the effects of automatic summarization of IGP-to-BGP redistribution are not desired

1. What is the AD of BGP routes in the IP routing table that were learned from BGP neighbors in a different AS? (Source: "Processing BGP Routes")

1

20

90

120

1. Which three BGP attributes are displayed for each route in the BGP table when you are using the **show ip bgp** command? (Choose three.) (Source: "Processing BGP Routes")

☒ weight

☐ communities

☒ origin

☒ AS-path

1. What is the valid AS number range for a BGP process on a Cisco router? (Source: "Configuring Basic BGP")

☐ 1 to 256

☐ 1 to 32768

☐ 1 to 65535

☐ 1 to 131072

1. Which two parameters must you configure with the **neighbor** command to establish a BGP session with an external neighbor? (Choose two.) (Source: "Configuring Basic BGP")

☒ neighbor IP address

☐ subnet mask of the IP network

☒ remote AS number

☐ local AS number

☐ description of the neighbor

1. What is the best method to temporarily disable a BGP neighbor session? (Source: "Configuring Basic BGP")

Remove the **neighbor** command from the BGP router process.

Remove the BGP router process from the configuration.

Terminate the neighbor connection with the **neighbor shutdown** command

Disconnect the neighbor by initiating a router reload.

1. Which two of the following statements about configuring BGP timers are accurate? (Choose two.) (Source: "Configuring Basic BGP")



Changing the BGP default holdtime and keepalive timers is usually not recommended.



The **neighbor timers** command sets the timers for a specific BGP peer or peer group.



The **timers bgp** command sets the timers for a specific BGP peer or peer group.



Holdtime indicates the frequency (in seconds) with which the Cisco IOS software sends messages to its peer.

1. Which two of the following are characteristics of the string component of the **neighbor** *{ip-address | peer-group-name}* **password** *string* command? (Choose two.) (Source: "Configuring Basic BGP")

- ☒ can contain any alphanumeric characters, including spaces
- ☐ case-sensitive password of up to 100 characters
- ☐ first character can be a number
- ☒ cannot specify a password in the format "number-space-anything"

1. Which three steps must you complete to advertise a classless prefix into BGP? (Choose three.) (Source: "Configuring Basic BGP")

- ☒ Configure the prefix with the **network** command.
- ☒ Specify the **mask** keyword with the locally advertised route.
- ☐ Configure the **redistribute connected** command under the BGP router process.
- ☒ Use a static route pointing to null 0 that matches the prefix.

1. Which origin code is carried with routes that are redistributed into BGP? (Source: "Configuring Basic BGP")

☐ internal

☐ external

☐ unknown

☐ incomplete

1. Which two of the following statements about the classless behavior of BGP are correct? (Choose two.) (Source: "Configuring Basic BGP")

☐ When an exact match is not found in the IP routing table a matching prefix is automatically configured on the router.

☐ In the **network** *ip-prefix-address* **mask** *subnet-mask* command, the prefix does not have to match an entry in the IP routing table.

☒ To advertise classless networks into BGP (a subnet or a supernet), you can use the **network** command with the **mask** keyword and the subnet mask specified.

☒ If the keyword **mask** and the subnet mask are omitted, the network is assumed to have its natural mask according to the network class.

1. What are two benefits of using route aggregation in BGP? (Choose two.) (Source: "Configuring Basic BGP")

☐ It ensures that even if aggregate networks are down, the aggregate is advertised, which eliminates black holes.

☒ It reduces the amount of memory that is used in the router to store the BGP table.

☒ It reduces route flapping and its effects on router CPU resources.

☐ BGP attribute granularity is maintained, which ensures optimal path selection.

1. Which two of the following are characteristics of the BGP Conditional Route Injection feature? (Choose two.) (Source: "Configuring Basic BGP")

☒ allows you to originate a prefix into a BGP routing table without the corresponding match

☒ enabled with the **bgp inject-map exist-map** command

☐ allows conditional injecting or replacing more specific prefixes with less specific prefixes

☐ allows origination of a prefix into a BGP routing table only with the corresponding match

1. Which two of the following are characteristics of the BGP Support for TTL Security Check feature? (Choose two.) (Source: "Configuring Basic BGP")

- ☐ should be configured on only one participating router
- ☐ prevents BGP sessions from expiring even if keepalive packets are not received before the session timer expires
- ☒ protects the EBGP peering session by comparing the value in the TTL field of received IP packets against a hop count that is configured locally for each EBGP peering session
- ☒ supports both directly connected peering sessions and multihop EBGP peering sessions

1. Which two of the following are functions of the **show ip bgp summary** command? (Choose two.)

- ☒ displays BGP memory use
- ☒ displays BGP neighbors and status of communication with them
- ☐ locates problems in BGP sessions that are up and established
- ☐ displays the BGP routing table

1. Which command do you use to display detailed BGP neighbor information? (Source: "Monitoring and Troubleshooting BGP")

☐ **show ip bgp summary**

☐ **show ip bgp**

☐ **show ip bgp neighbors address**

☐ **show ip bgp detail**

1. To get information about the TCP session and BGP parameters of the session, you can issue **show ip bgp summary** command. True or false? (Source: "Monitoring and Troubleshooting BGP")

true

false

1. Which two of the following statements about the **show ip bgp** command that is used to monitor the BGP routing table are accurate? (Choose two.) (Source: "Monitoring and Troubleshooting BGP")

☐ The **show ip bgp** command shows an abbreviated list of information about each route, displaying one line per prefix.

☒ The **show ip bgp** command shows a full list of information about each route, displaying one line per prefix.

☐ All of the BGP attributes that are associated with the route are displayed on the line.

☒ If the BGP table contains more than one route to the same network, the routes are displayed on successive lines of the command output.

1. Which debug command should you enable to troubleshoot BGP session startup issues where the TCP connection never succeeds? (Source: "Monitoring and Troubleshooting BGP")

☐ **ip bgp updates**

☐ **ip packets**

☐ **ip bgp keepalives**

☒ **ip tcp transactions**

1. When debugging an exchange of updates, you should use **debug ip bgp keepalive** command. True or False?

true

false

1. What are the three most common session startup issues that you can experience when configuring basic BGP? (Choose three.)
(Source: "Monitoring and Troubleshooting BGP")

☒	BGP neighbors do not become active.
☐	BGP routing loops cause black holes.
☒	A BGP neighbor is active, but the BGP session is not established.
☒	The BGP neighbor state oscillates between Idle and Active.
☐	The BGP session is active, but the neighbor cannot be reached.
☐	BGP keepalives experience intermittent failures.

1. What is the most common reason for a BGP session not leaving the Idle state? (Source: "Monitoring and Troubleshooting BGP")

- ☐ The TCP port for the connection is not configured.
- ☐ The external neighbor is not directly connected.
- ☐ The TCP SYN packet is answered with an RST packet.
- ☐ The neighbors have been configured with the same AS number.

1. What will result from attempting to open a BGP connection with a neighbor that has not been properly configured for BGP? (Source: "Monitoring and Troubleshooting BGP")

The BGP session will remain in the Idle state.

The neighbor session will be established, and the session startup parameters will be negotiated over the TCP session.

The BGP session will be immediately terminated with a TCP RST packet.

The BGP session will become "stuck in Active state."

1. When a BGP neighbor oscillates between Active and Idle, what is the likely diagnosis? (Source: "Monitoring and Troubleshooting BGP")

There are mismatched keepalive intervals.

There is an AS number mismatch between BGP neighbors.

One router is not configured as the BGP neighbor on the neighboring router.

The BGP neighbor is not reachable.

BGP Transit Autonomous Systems

Introduction

This module is one of the focal points of the [BGP](#) curriculum: a discussion of BGP issues in a transit [AS](#). In this module you will learn basic BGP transit AS issues, ranging from synchronization between an [IGP](#), and BGP to [IBGP](#) full-mesh and next-hop requirements.

Upon completing this module, you will be able to:

- Describe the function of a transit AS and the need for IBGP
- Describe the interaction in a transit AS between EBGP and IBGP
- Describe the function of an IGP in forwarding packets through an AS
- Verify proper operation of a configured BGP transit network by performing the steps necessary to correct basic IBGP configuration errors

Working with Transit AS

Overview

All transit autonomous systems are required to carry traffic originating from or destined for locations outside of that [AS](#). For the transit AS to meet this requirement, a degree of interaction and coordination between [BGP](#) and the [IGP](#) that this particular AS uses is necessary. Such a configuration requires special care to ensure consistency of routing information throughout the AS.

The topology of the Internet can be viewed as a series of connections between stub networks, multihomed networks, and transit autonomous systems. A multihomed AS containing more than one connection to the outside world and allowing traffic not originating in that AS to travel through it is a transit AS. This lesson introduces the concept of the multihomed transit AS and how BGP exchanges routing information inside the AS and between neighboring autonomous systems. It also explains the requirement for [IBGP](#) within the multihomed transit AS.

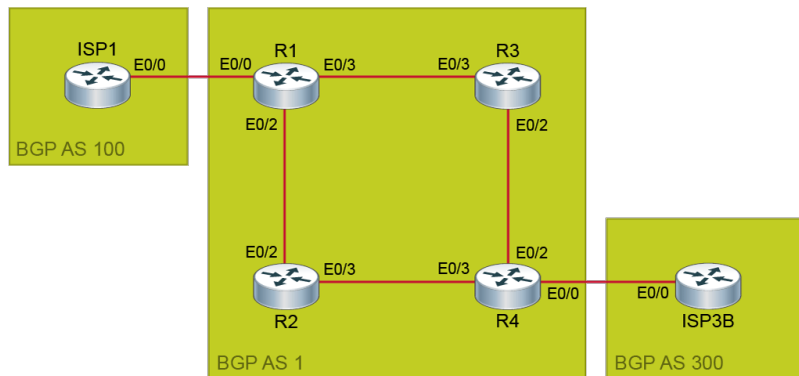
Upon completing this lesson, you will be able to:

- List the functions of a transit AS
- Describe external route propagation between autonomous systems in a BGP network
- Describe internal route propagation within a BGP AS
- Explain how transiting packets are forwarded inside a transit AS
- Explain the need for deploying IBGP on all core routers

Transit AS Tasks

A transit [AS](#) is required to carry traffic that originates from or is destined for locations outside of that AS. A transit AS has two basic functions.

- Propagate routes between remote autonomous systems
- Route packets between remote networks

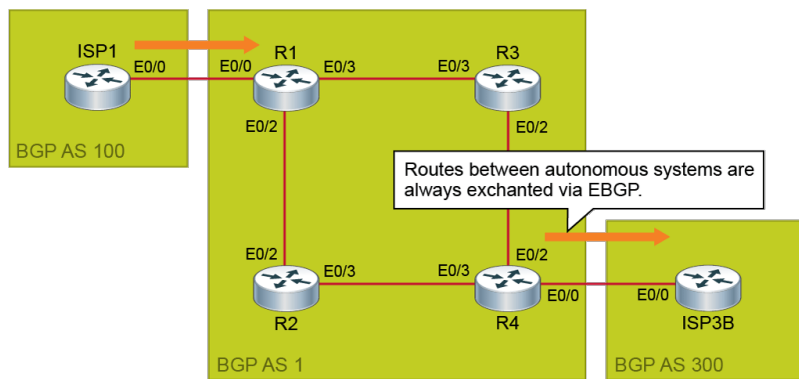


Routers in a transit AS have to perform two tasks:

- Receive routing information updates about reachable networks from neighboring autonomous systems, propagate the information through their own AS, and send it to other neighboring autonomous systems.
- Forward [IP](#) packets that they have received from a neighboring AS through their own AS to a downstream neighboring AS. The routers in the transit AS perform this task using the routing information that they have received as part of the first task.

External Route Propagation

Two autonomous systems usually exchange routing information about reachable networks using [BGP](#). There is currently no alternative routing protocol that has the scalability and security characteristics of BGP.



In the figure, the BGP session between ISP1 and R1 is called an [EBGP](#) session because ISP1 and R1 are in different autonomous systems.

BGP routing information updates consist of the network address, subnet mask, and any number of BGP attributes. No other routing protocol provides the same richness of route attributes as BGP. Translating BGP route attribute information into any other protocol would likely cause a loss of information. Therefore, the EBGP information that R1 receives is not translated; it is just forwarded to other BGP-speaking routers (R2, R3, and R4 in the figure) within the [AS](#).

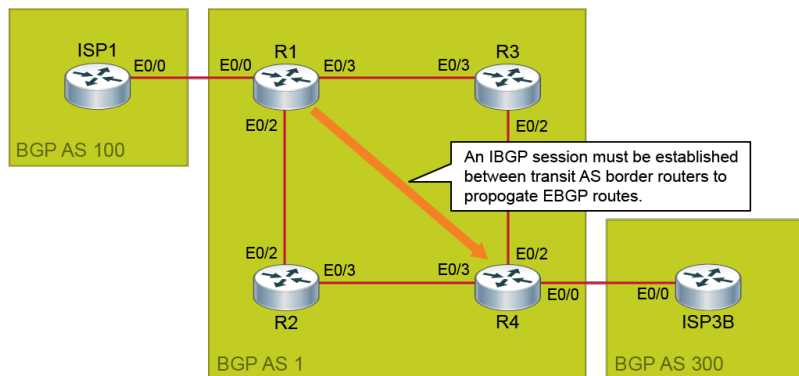
Likewise, R4 has BGP information and can propagate it to ISP3B in AS 300 over the EBGP session.

EBGP sessions are, in general, established between directly connected neighbors. BGP-speaking routers, therefore, need no additional routing information to establish a session.

Internal Route Propagation

In this example, the BGP session between R1 and R4, which are both in the same AS, is an IBGP session.

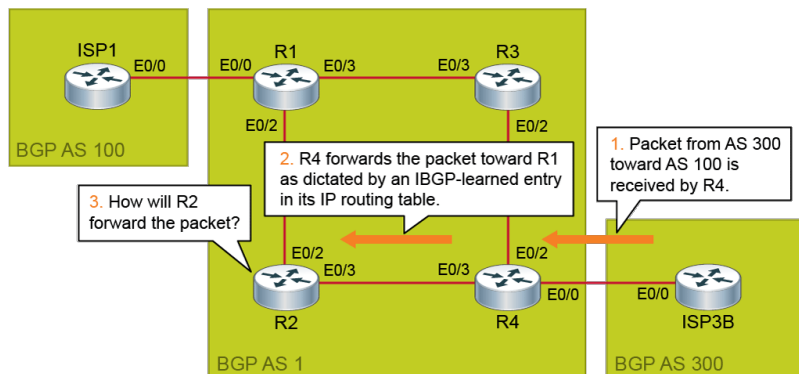
The only protocol that can transport all BGP attributes across the backbone is BGP inside an AS, called Internal BGP (IBGP).



IBGP sessions are, in general, established between distant routers in the same AS. These routers need extra routing information to establish the session, because there is no requirement that IBGP neighbors be directly connected. This information typically comes from the IGP, which is running within the AS independently of BGP.

Packet Forwarding in AS

In this example, after [AS 300](#) has received the routing information about reachable networks inside [AS 100](#), [IP](#) packets can start to flow. In the figure, IP packets flow from AS 300 toward AS 100. ISP3B, the egress router in AS 300, forwards IP packets with destinations in AS 100 toward R4, according to information received through [EBGP](#).



Conclusions:

- R2 needs external routes for proper packet forwarding. R2 must receive BGP routes.

R4 now uses the [IBGP](#) information that it received from R1 and forwards the packets in the direction of R1, which in this case means via R2.

When the IP packets reach R2, the router checks its routing table for a matching entry, but it fails to find one. The packet is dropped because the destination network is unreachable from the perspective of R2.

This situation is, of course, unacceptable. To prevent dropped packets resulting from unreachable networks, R2 must also have routing information about the networks reachable inside AS 100. The same information that R4 received from R1 over the IBGP session must be propagated to R2.

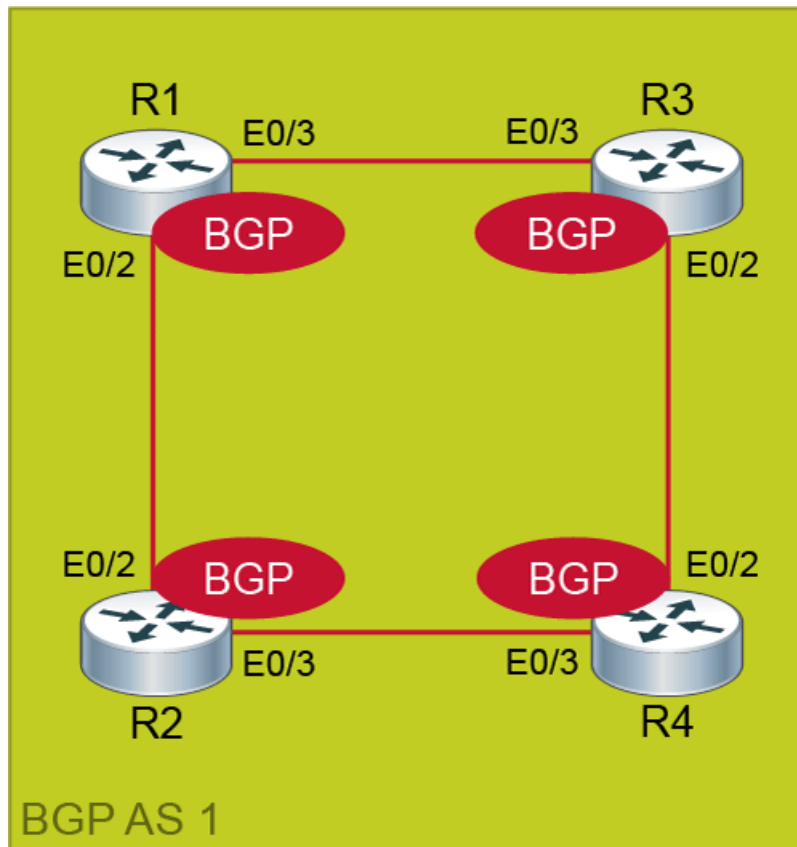


R3 has the same network reachability requirements as R2, because R4 could forward the packets via R3 as well as via R2.

Core Router IBGP Requirements in Transit AS

Within a transit [AS](#), all routers that are in a theoretical transit path between external destinations should have information about all external routes that are received from any neighboring AS. If a single router on a transit path does not have this information, there is always a possibility that an [IP](#) packet that is received from a neighboring AS will not be able to be forwarded by that router through the transit AS. The router lacking routing information about the final destination of the IP packet drops it into what effectively becomes a black hole.

- All core routers must have all external routes.
- Core routers must receive BGP routes.
 - Redistribution of BGP routes into IGP is not scalable.
 - Default routing is not applicable in transit AS core.



The only feasible way for the router to distribute all external routing information is by using [BGP](#). Redistribution of the [EBGP](#) routes into an [IGP](#) is not viable because no IGP can carry the volume of information that BGP currently carries in the Internet.



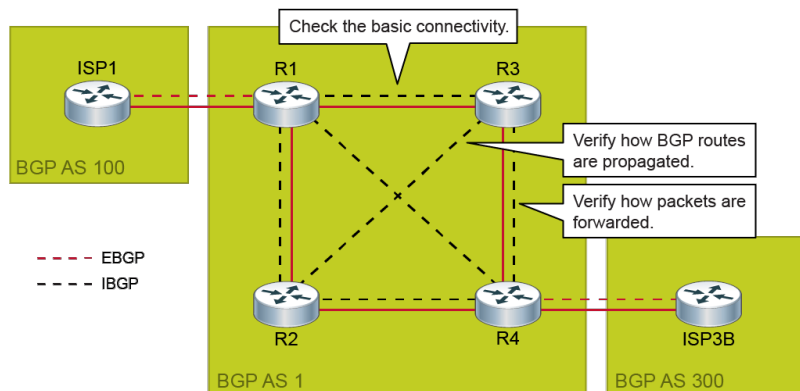
The risk of losing information during redistribution of EBGP routes into an IGP is not the reason why BGP is used to update intermediate routers in the transit path instead of an IGP. Redistribution into an IGP is not used because of the scalability issues that would arise from doing so.

Default routing or a gateway of last resort cannot be used by routers within the transit path when transit services are provided to other autonomous systems. If some routes are filtered out and the default route is used instead, full routing flexibility is lost. In this case, the transit AS is not able to forward packets to all destinations at all times. In fact, routing loops and black holes might be easily introduced.

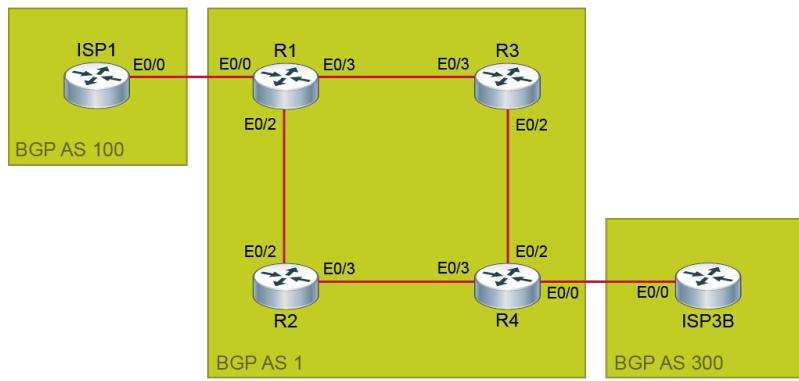
Discovery 4: BGP Route Propagation

Overview

Through this discovery, you will learn how the [BGP](#) routes are propagated between different autonomous systems that are connected directly and need for full mesh within an [AS](#).



Topology



Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
ISP1	Ethernet 0/0	172.16.11.1/24	Connection to R1
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R1	Ethernet 0/3	192.168.13.1/24	Connection to R3
R1	Loopback 1	10.0.0.1/28	Loopback simulates LAN network
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Ethernet 0/3	192.168.24.2/24	Connection to R4
R2	Loopback 1	10.0.0.33/28	Loopback simulates LAN network
R3	Ethernet 0/2	192.168.34.3/24	Connection to R4
R3	Ethernet 0/3	192.168.13.3/24	Connection to R1
R3	Loopback 1	10.0.0.17/28	Loopback simulates LAN network
R4	Ethernet 0/0	172.16.34.4/24	Connection to ISP3B
R4	Ethernet 0/2	192.168.34.4/24	Connection to R3
R4	Ethernet 0/3	192.168.24.4/24	Connection to R2
R4	Loopback 1	10.0.0.49/28	Loopback simulates LAN network
ISP3B	Ethernet 0/0	172.16.34.34/24	Connection to R4
ISP3B	Loopback 1 Loopback 2 Loopback 3 Loopback 4	10.0.3.1/28 10.0.3.17/28 10.0.3.33/28 10.0.3.49/28	Loopbacks simulate LAN networks

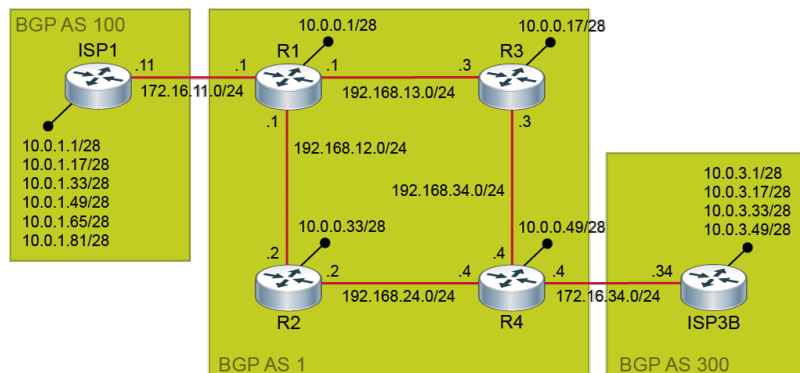
BGP Route Propagation

Discovery Steps

Step 1

You will verify route 10.0.1.0/28 is propagated from AS 100 into AS 300.

IP addresses, IGP and BGP are preconfigured as shown in the topology below:



On the ISP1 router, verify route 10.0.1.0/28 is present in the IP routing table.

```
ISP1# show ip route
< output omitted >
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 20 subnets, 2 masks
B       10.0.0.0/28 [20/0] via 172.16.11.1, 01:49:30
B       10.0.0.16/28 [20/0] via 172.16.11.1, 01:49:30
B       10.0.0.32/28 [20/0] via 172.16.11.1, 01:49:30
B       10.0.0.48/28 [20/0] via 172.16.11.1, 01:49:30
C       10.0.1.0/28 is directly connected, Loopback1
L       10.0.1.1/32 is directly connected, Loopback1
C       10.0.1.16/28 is directly connected, Loopback2
L       10.0.1.17/32 is directly connected, Loopback2
C       10.0.1.32/28 is directly connected, Loopback3
<... output omitted ...>
```

IP address 10.0.1.1/28 is configured on the Loopback 1 interface.

Step 2

On the ISP1 router, verify route 10.0.1.0/28 is present in the BGP routing table as locally originated.

```
ISP1# show ip bgp
< output omitted >

   Network          Next Hop        Metric LocPrf Weight Path
*>  10.0.0.0/28      172.16.11.1            0                0 1 i
*>  10.0.0.16/28     172.16.11.1            0                0 1 i
*>  10.0.0.32/28     172.16.11.1            0                0 1 i
*>  10.0.0.48/28     172.16.11.1            0                0 1 i
*>  10.0.1.0/28      0.0.0.0                0          32768 i
*>  10.0.1.16/28     0.0.0.0                0          32768 i
*>  10.0.1.32/28     0.0.0.0                0          32768 i
*>  10.0.1.48/28     0.0.0.0                0          32768 i
*>  10.0.1.64/28     0.0.0.0                0          32768 i
*>  10.0.1.80/28     0.0.0.0                0          32768 i
*>  10.0.3.0/28      172.16.11.1            0                0 1 300 i
*>  10.0.3.16/28     172.16.11.1            0                0 1 300 i
*>  10.0.3.32/28     172.16.11.1            0                0 1 300 i
*>  10.0.3.48/28     172.16.11.1            0                0 1 300 i
```

The route 10.0.1.0/28 is locally originated, because next hop IP address is 0.0.0.0 and AS path attribute is empty.

Step 3

On the R1 router, verify route 10.0.1.0/28 is present in the BGP routing table.

```
R1# show ip bgp
< output omitted >
      Network                Next Hop                Metric LocPrf Weight Path
*> 10.0.0.0/28                0.0.0.0                    0           32768 i
*>i 10.0.0.16/28              10.0.0.17                   0          100      0 i
*>i 10.0.0.32/28              10.0.0.33                   0          100      0 i
*>i 10.0.0.48/28              10.0.0.49                   0          100      0 i
*> 10.0.1.0/28                172.16.11.11                0           0 100 i
*> 10.0.1.16/28               172.16.11.11                0           0 100 i
*> 10.0.1.32/28               172.16.11.11                0           0 100 i
*> 10.0.1.48/28               172.16.11.11                0           0 100 i
*> 10.0.1.64/28               172.16.11.11                0           0 100 i
*> 10.0.1.80/28               172.16.11.11                0           0 100 i
*>i 10.0.3.0/28               172.16.34.34                0          100      0 300 i
*>i 10.0.3.16/28              172.16.34.34                0          100      0 300 i
*>i 10.0.3.32/28              172.16.34.34                0          100      0 300 i
*>i 10.0.3.48/28              172.16.34.34                0          100      0 300 i
```

The route 10.0.1.0/28 has next hop IP address 172.16.11.11 and AS path attribute 100.

Step 4

On the R4 router, verify route 10.0.1.0/28 is present in the BGP routing table.

```
R4# show ip bgp
< text omitted >
      Network                Next Hop                Metric LocPrf Weight Path
*>i 10.0.0.0/28                10.0.0.1                    0          100      0 i
*>i 10.0.0.16/28              10.0.0.17                   0          100      0 i
*>i 10.0.0.32/28              10.0.0.33                   0          100      0 i
*> 10.0.0.48/28                0.0.0.0                    0           32768 i
*>i 10.0.1.0/28                172.16.11.11                0          100      0 100 i
*>i 10.0.1.16/28               172.16.11.11                0          100      0 100 i
*>i 10.0.1.32/28               172.16.11.11                0          100      0 100 i
*>i 10.0.1.48/28               172.16.11.11                0          100      0 100 i
*>i 10.0.1.64/28               172.16.11.11                0          100      0 100 i
*>i 10.0.1.80/28               172.16.11.11                0          100      0 100 i
*> 10.0.3.0/28                 172.16.34.34                0           0 300 i
*> 10.0.3.16/28                172.16.34.34                0           0 300 i
*> 10.0.3.32/28                172.16.34.34                0           0 300 i
*> 10.0.3.48/28                172.16.34.34                0           0 300 i
```

The route 10.0.1.0/28 has next hop IP address 172.16.11.11 and AS path attribute 100. You can see route is learned via internal BGP.

Step 5

On the R4 router, verify what is the best path to reach route next hop IP address 172.16.11.11.

```
R4# show ip route
< output omitted >
      172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
O          172.16.11.0/24 [110/30] via 192.168.34.3, 02:13:15, Ethernet0/2
          [110/30] via 192.168.24.2, 02:13:15, Ethernet0/3
<... output omitted ...>
```

Output shows that next hop IP address 172.16.11.11 is reachable via two [OSPF](#) routes. Router R4 will load share between two OSPF routes. Some packets will be sent to the R2 router and some to the R3 router.

So, R2 and R3 routers need IP routing information to reach next hop 172.16.11.11.

Step 6

On the ISP3B router, verify route 10.0.1.0/28 is present in the BGP routing table.

```
ISP3B# show ip bgp
< output omitted >
      Network                Next Hop                Metric LocPrf Weight Path
```

*>	10.0.0.0/28	172.16.34.4		0	1	i
*>	10.0.0.16/28	172.16.34.4		0	1	i
*>	10.0.0.32/28	172.16.34.4		0	1	i
*>	10.0.0.48/28	172.16.34.4	0	0	1	i
*>	10.0.1.0/28	172.16.34.4		0	1	100 i
*>	10.0.1.16/28	172.16.34.4		0	1	100 i
*>	10.0.1.32/28	172.16.34.4		0	1	100 i
*>	10.0.1.48/28	172.16.34.4		0	1	100 i
*>	10.0.1.64/28	172.16.34.4		0	1	100 i
*>	10.0.1.80/28	172.16.34.4		0	1	100 i
*>	10.0.3.0/28	0.0.0.0	0	32768		i
*>	10.0.3.16/28	0.0.0.0	0	32768		i
*>	10.0.3.32/28	0.0.0.0	0	32768		i
*>	10.0.3.48/28	0.0.0.0	0	32768		i

The route 10.0.1.0/28 has next hop IP address 172.16.34.4 and AS path attribute 1 100.

Step 7

You will verify, how packets are forwarded from AS 300 to the IP 10.0.1.1 in the AS 100. On the ISP3B router, ping next hop IP address of the BGP route 10.0.1.0/24.

In the previous step, you have learned next hop IP address is 172.16.34.4. Source ping from the ISP3B router's Loopback 1 interface.

```
ISP3B# ping 172.16.34.4 source Loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.34.4, timeout is 2 seconds:
Packet sent with a source address of 10.0.3.1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

From the ISP3B router, you can successfully reach next hop 172.16.34.4 (router R4).

Step 8

From the R4 router, to reach IP address 10.0.1.1, next hop IP address is 172.16.11.11. From the R4 router, ping next hop IP address from the BGP route 10.0.1.0/24.

Source ping from the R4 router's Loopback 1 interface.

```
R4# ping 172.16.11.11 source Loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.11.11, timeout is 2 seconds:
Packet sent with a source address of 10.0.0.49
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

From the R4 router, you can successfully reach next hop 172.16.11.11 (ISP1 router).

Step 9

From the ISP3B router, ping directly to the IP address 10.0.1.1. Source ping from the ISP3B router's Loopback 1 interface.

Ping should be successful. The successful ping demonstrates IP connectivity between AS 100 and AS 300 via AS 1.

```
ISP3B# ping 10.0.1.1 source Loopback 1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.3.1
!!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

Summary

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

- Routers in a transit AS receive routing information updates from neighboring autonomous systems, propagate the information through their own AS, and send it to other neighboring autonomous systems.
- Two autonomous systems usually exchange routing information over an EBGP session.
- A BGP session between two routers in the same AS is called an IBGP session.
- For packets to be properly forwarded in a transit AS, all routers must have external routing information.
- The only feasible method of distributing external routing information to all routers in the transit AS is through IBGP.

Interacting with IBGP and EBGP in Transit AS

Overview

Configuring a [BGP](#) network in a transit services configuration requires special care to ensure consistency of routing information throughout the [AS](#). Understanding the interaction between [EBGP](#) and [IBGP](#) is crucial to successfully configuring and troubleshooting the transit autonomous network.

Upon completing this lesson, you will be able to:

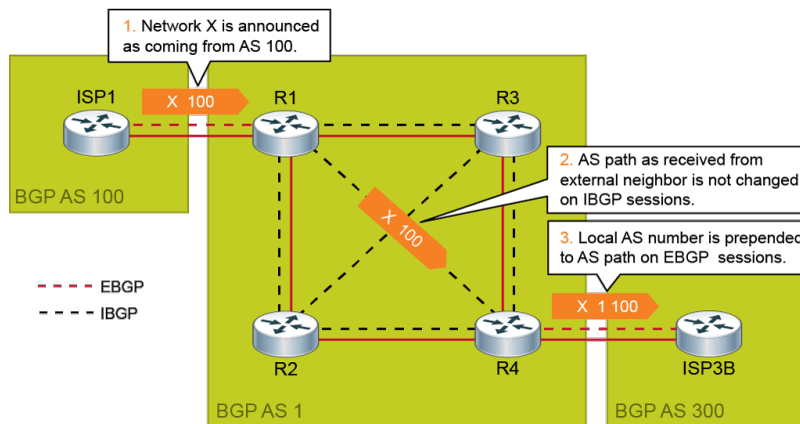
- Describe AS-path processing in IBGP
- Describe BGP multipath load sharing
- Explain the need for BGP split horizon
- Explain the need for a full-mesh topology between IBGP routers and the implications of that need
- List the benefits of establishing IBGP neighbor sessions using loopback interfaces
- Describe next-hop processing in IBGP
- Explain why all EBGP peers must be reachable by all BGP-speaking routers within the AS
- Describe how to configure edge routers to announce themselves as the next hop in IBGP updates
- Describe the differences between EBGP and IBGP sessions
- List the scalability limitations of IBGP-based backbones

AS-Path Processing in IBGP

All BGP routing updates carry the mandatory well-known attribute AS-path, which lists the autonomous systems that the routing update has already crossed.

When a router originates a BGP prefix (network X in this example), the AS path is empty. Whenever a BGP prefix is announced over an EBGP session, the AS number of the router that is sending the information is prepended to the AS path. In the example, ISP1 inserts "100" in the AS path before forwarding the routing update to R1.

The AS path is not changed when the BGP prefix is propagated across IBGP sessions because the routing update has not crossed an AS boundary.

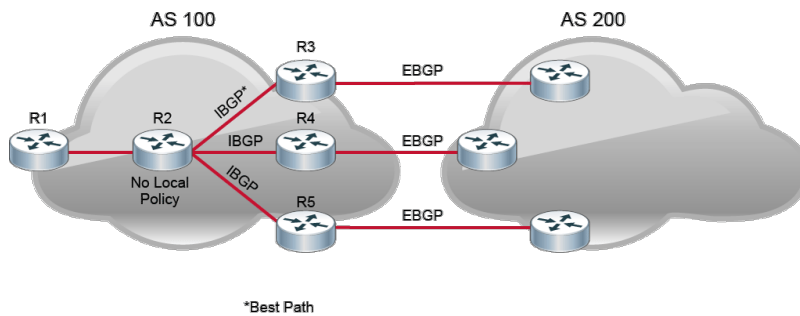


In the figure, R1 forwards the information over an IBGP session to R4 with the AS-path unchanged. The AS-path information about network X will be the same in all routers within AS 1, because all the routers are updated using IBGP sessions from R1.

When R4 forwards the information about network X to ISP3B, R4 prepends its own AS number (1) to the AS path. Thus, ISP3B receives the routing information about network X with an AS-path attribute of "1 100."

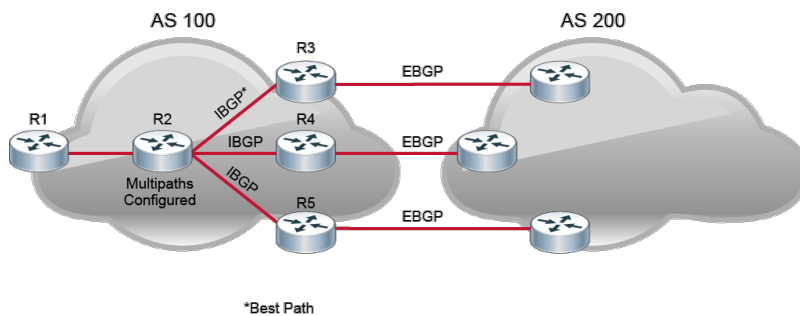
Multipath Load Sharing in BGP

When a BGP-speaking router with no local policy configured receives NLRI from multiple IBGP sources for the same destination, the router chooses one IBGP path as the best path. The best path is then installed in the IP routing table of the router.



The figure illustrates that with three paths to AS 200, the R2 router determines that one of the paths to AS 200 is the best path. So, it uses only this path to reach AS 200.

The IBGP multipath load-sharing feature enables the BGP-speaking router to select multiple IBGP paths as the best paths to a destination. The best paths, or multipaths, are then installed in the IP routing table of the router.



On the R2 router in the figure, the paths to the R3, R4, and R5 routers are configured as multipaths and can be used to reach AS 200. The load to AS 200 is equally shared between the paths.

For multiple paths to the same destination to be considered as multipaths, the following criteria must be met:

- All attributes must be the same. The attributes include weight, local preference, AS path (entire attribute and not just length), origin code, MED, and IGP distance.
- The next hop router for each multipath must be different.

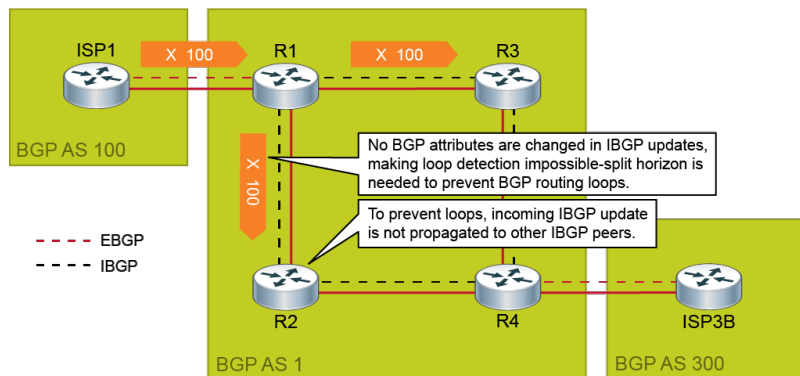
Even if the criteria are met and multiple paths are considered multipaths, the BGP-speaking router still designates one of the multipaths as the best path and advertises this best path to its neighbors.

Configuring multiple IBGP best paths enables a router to evenly share the traffic that is destined for a particular site.

Use the **maximum-paths ibgp** command in router configuration mode, to control the maximum number of parallel internal BGP routes that can be installed in a routing table.

BGP Split Horizon

All routers within an [AS](#) must make routing decisions in a consistent way. They must have access to the same routing information with the same attributes in order to come to the same conclusion about which exit point of the AS to use. In other words, the [BGP](#) attributes should not be changed within the AS.



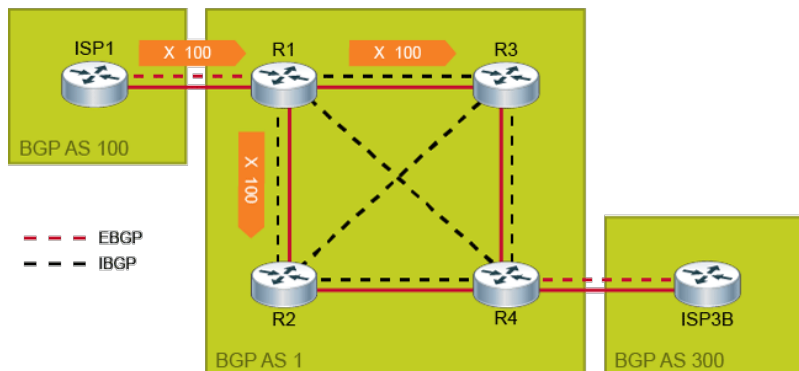
Result: Full mesh of IBGP sessions is required for proper IBGP update propagation.

The AS-path attribute is not changed over an [IBGP](#) session, because the BGP update has not crossed the AS boundary. However, the AS-path attribute is the primary means of detecting routing information loops. A BGP router that encounters its own AS in the AS path of an incoming BGP update silently ignores the information. Because the BGP-speaking routers modify AS path only on [EBGP](#) sessions, this loop-preventing mechanism is only useful between autonomous systems, not within them.

IBGP split horizon prevents routing information loops within the AS. Routing information that is received through an IBGP session is never forwarded to another IBGP neighbor, only toward EBGP neighbors. Because of BGP split horizon, no router can relay IBGP information within the AS—all routers must be directly updated from the border router that received the EBGP update.

IBGP Full Mesh

- Full mesh of IBGP sessions has to be established between all BGP-speaking routers in the AS for proper IBGP route propagation.
- The IBGP full mesh is a logical mesh of TCP sessions only; physical full mesh is not required.



Every router on the transit path within the [AS](#) must have routing information about all external networks that are received by any of the border routers. So, R2 and R3 must have [IBGP](#) sessions to all border routers.

This level of communication is not enough, though, because any of the internal routers could also create new [BGP](#) routing information (for example, originate a customer network). These updates must also reach all the routers within the AS. The conclusion is that all BGP routers within an AS must have IBGP sessions with every other BGP router in the AS. This requirement results in a full mesh of BGP sessions between BGP-speaking routers in an AS.

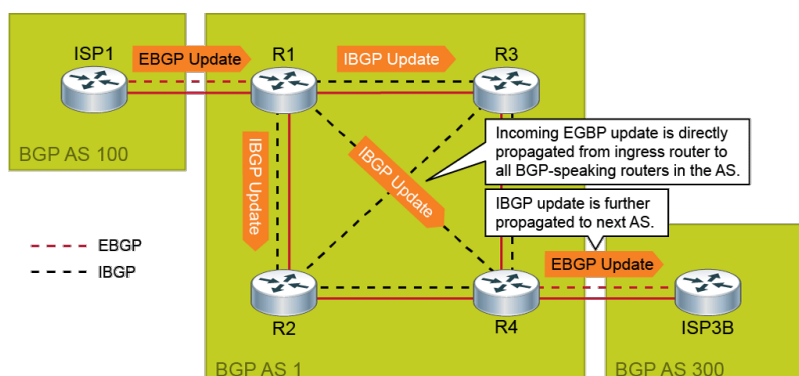
In the network in the figure, R1 must have IBGP sessions with R2, R3, and R4 to propagate routes that are received from AS 100 to all routers within AS 1. Similarly, R4 must have IBGP sessions with R1, R2, and R3 to be able to propagate routes that are received from AS 300 to all routers within AS 1.

i The IBGP session between R2 and R3 is not strictly necessary for proper forwarding of [IP](#) packets between external destinations. It does become mandatory if R2 or R3 starts to originate BGP networks. To prevent potential future connectivity issues, it is a good practice to establish a full mesh of IBGP sessions regardless of whether they are needed at the time of network deployment or not.

The [IGP](#) that runs within AS 1 provides enough information to any BGP router within AS 1 to send IP packets to any other router in the AS. Having enough router reachability information makes it possible to establish IBGP sessions between routers even though they are not physically connected. The IBGP full mesh is a logical full mesh of [TCP](#) sessions and will run on an arbitrary physical topology.

IBGP Full Mesh Example

The figure illustrates IBGP split-horizon and IBGP full-mesh principles in a sample network.



ISP1 is sending an update to R1 over an EBGP session. Updates that are received on an [EBGP](#) session should be forwarded on all other IBGP sessions, so R1 updates R2, R3, and R4. All routers within AS 1 are updated directly by R1.

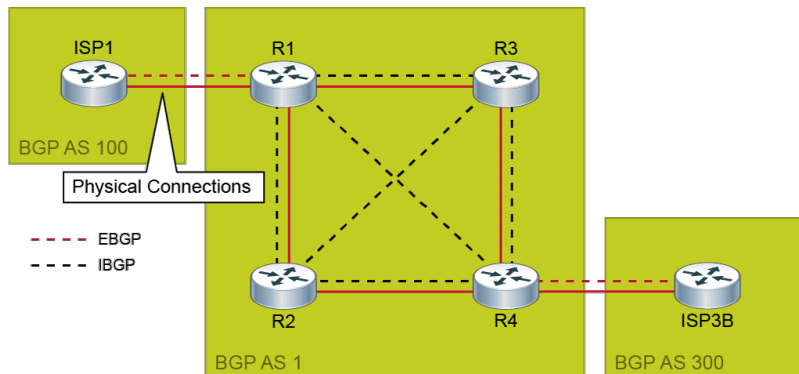
R2 and R3 are prevented from forwarding the update that they received from R1 because of BGP split horizon.

R4, which received the information on an IBGP session, is prevented from updating R2 and R3 because of the same split-horizon rule. But R4 will update ISP3B over an EBGP session.

IBGP Neighbors

In the figure, the transit AS 1 has a redundant physical topology. The IGP provides reachability information for all routers and networks within AS 1. This way, IGP enables all routers in the AS to establish IBGP sessions to all other routers, even if the routers are not directly connected.

Because of IBGP full-mesh requirements, IBGP neighbors are usually not directly connected. Which interfaces should be chosen as the source and destination addresses of IBGP TCP sessions?



If the IBGP session between R1 and R4 was established using IP addresses that belong to the physical interfaces, the IBGP session would go down if either of the physical interfaces went down. The IP address of an interface that is in the down state is invalid. As a result, the router would tear down the TCP session that is used for BGP between the routers. Then, all IP packets that are received with a destination address pointing to that interface will also be dropped.

During the network design and implementation phase, the network designers must be careful that those IBGP sessions remain established for as long as the two BGP routers have any usable path between them.

Always run IBGP sessions between loopback interfaces.

- IBGP sessions can always be established, even if some physical interfaces are down.
- IBGP sessions are stable—physical interface failure will not tear down IBGP sessions.
- There is no BGP recovery after a failure inside the transit AS.
 - The configured IGP will re-establish the path between loopback interfaces.
 - IBGP sessions are not affected.

The best choice when you are configuring IBGP sessions is to establish each session between loopback interfaces on each BGP router.

To establish BGP connectivity between the loopback interfaces, the IP addresses of these interfaces have to be reachable by both routers. It is important that the IGP carry information about the subnets that are assigned to each loopback interface so that the interfaces are reachable by all BGP routers in the AS.

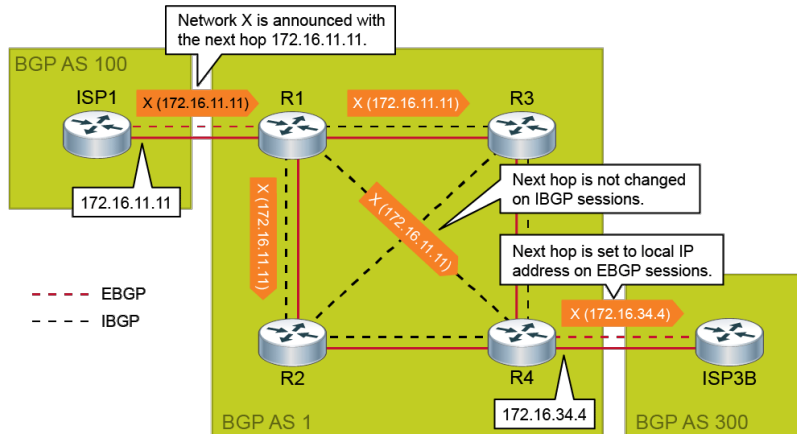
The IBGP sessions that are established between loopback interfaces have increased stability. These sessions will not go down if a single physical interface goes down. As long as the IGP can find any path between the two routers, the IBGP session will remain up. BGP will not notice that the IGP has changed the traffic path between the two routers.



Because BGP sessions run over TCP, they can survive even a short loss of connectivity between BGP routers with no impact to the BGP routing protocol. The only requirement placed on the IGP is that the network must converge before the BGP keepalive timer expires.

IBGP Next-Hop Processing

Every BGP update carries the mandatory well-known attribute next hop. The next-hop attribute specifies the IP address that the router should use as the forwarding next hop for packets that are sent toward the announced destination address. In most cases, the next hop is set to the IP address that the sending router is using as its source IP address for EBGP sessions. The receiving BGP router will use the information and route IP packets toward the announced destination via the indicated next hop, which is normally directly connected.



The next-hop attribute is not changed on IBGP updates. So, when the border router forwards the BGP update on IBGP sessions, the next-hop address is still set to the IP address of the far end of the EBGP session. Therefore, the receiver of IBGP updates will see the next-hop information indicating a destination that is not directly connected. To resolve this problem, the router will check its routing table and see if and how it can reach the next-hop address. The router can then route IP packets with destination addresses matching the network in the BGP update in the same direction as it would have routed an IP packet with a destination address equal to the IP address stated in the next-hop attribute. This process is known as recursive routing.

In the figure, ISP1 sends a BGP update about network X. Because it is sending this update over an EBGP session to R1, the next-hop attribute is set to the IP address that is used at the ISP1 side of the EBGP session, 172.16.11.11.

R1 can use this information and route packets to network X by forwarding them to ISP1.

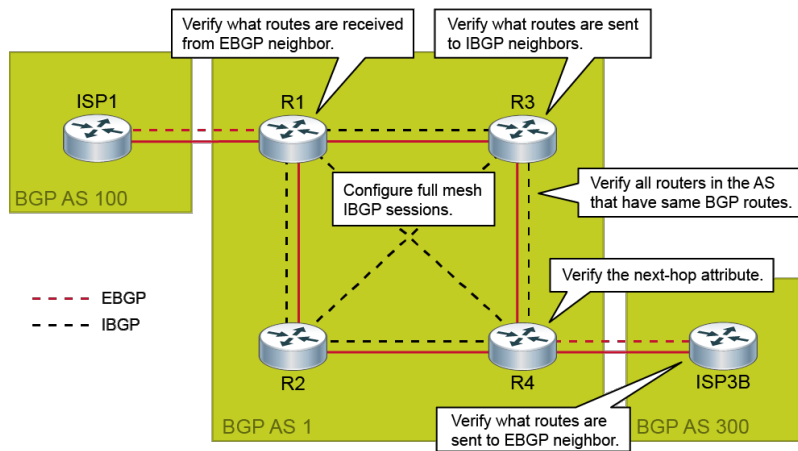
R1 also forwards the BGP update over all its IBGP sessions. It does not change the next-hop attribute, so R2, R3, and R4 get information that they can reach network X by forwarding packets to 172.16.11.11. But that IP address is not directly connected, so the routers must look in their routing tables to see if and how they can reach 172.16.11.11. If the recursive route lookup is successful, each router can then route packets to network X in the same direction as it would route packets to 172.16.11.11.

R4 also forwards the BGP update about network X to ISP3B. The connection between these routers is an EBGP session. So, R4 sets the next-hop attribute to its own IP address, 172.16.34.4, which is used by R4 on the EBGP session toward ISP3B.

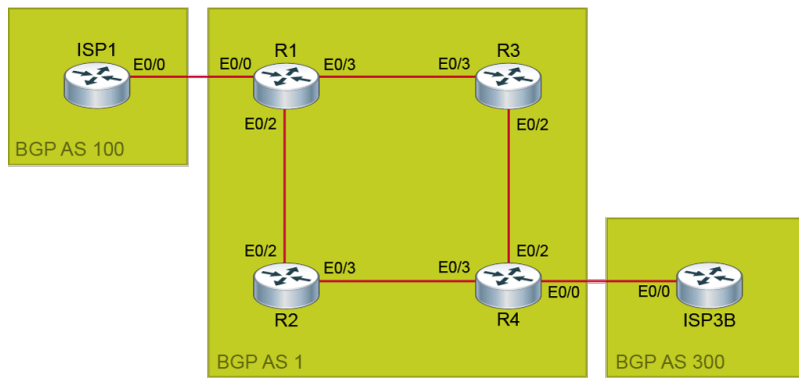
Discovery 5: IBGP Full Mesh

Overview

Through this discovery, you will learn how the [BGP](#) split horizon works and how it impacts the routing table. You will learn why full mesh [IBGP](#) sessions within an [AS](#) are required and how next-hop is processed during route propagation within an AS.



Topology



Job Aids

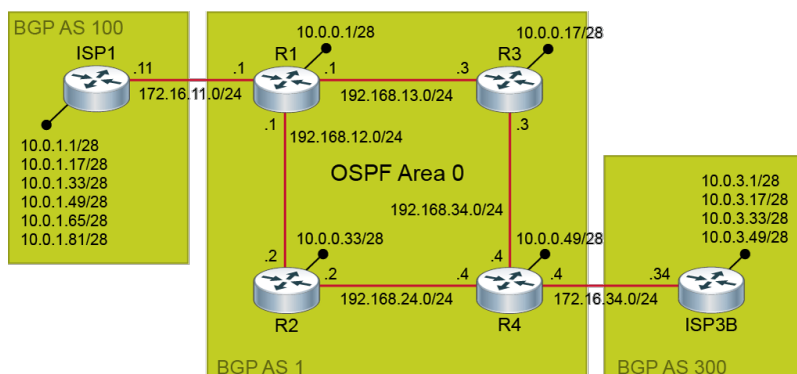


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
ISP1	Ethernet 0/0	172.16.11.1/24	Connection to R1
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R1	Ethernet 0/3	192.168.13.1/24	Connection to R3
R1	Loopback 1	10.0.0.1/28	Loopback simulates LAN network
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Ethernet 0/3	192.168.24.2/24	Connection to R4
R2	Loopback 1	10.0.0.33/28	Loopback simulates LAN network
R3	Ethernet 0/2	192.168.34.3/24	Connection to R4
R3	Ethernet 0/3	192.168.13.3/24	Connection to R1
R3	Loopback 1	10.0.0.17/28	Loopback simulates LAN network
R4	Ethernet 0/0	172.16.34.4/24	Connection to ISP3B
R4	Ethernet 0/2	192.168.34.4/24	Connection to R3
R4	Ethernet 0/3	192.168.24.4/24	Connection to R2
R4	Loopback 1	10.0.0.49/28	Loopback simulates LAN network
ISP3B	Ethernet 0/0	172.16.34.34/24	Connection to R4
ISP3B	Loopback 1 Loopback 2 Loopback 3 Loopback 4	10.0.3.1/28 10.0.3.17/28 10.0.3.33/28 10.0.3.49/28	Loopbacks simulate LAN networks

IP addresses and IGP are preconfigured as shown in the topology below:



BGP is also configured as [EBGP](#) (ISP1 to R1 and ISP3B to R4) and as IBGP (R1 to R2, R1 to R3, R3 to R4, and R2 to R4). IBGP is not configured between R1 to R4 or between R2 to R3.

IBGP Full Mesh

Discovery Steps

Step 1

On the R1 router, verify what BGP routes are received from ISP1 router.

An external BGP session is established between R1 and ISP1 routers.

```
R1# show ip bgp
BGP table version is 10, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  10.0.0.0/28      0.0.0.0              0         32768 i
*>i 10.0.0.16/28      10.0.0.17             0         100   0 i
*>i 10.0.0.32/28      10.0.0.33             0         100   0 i
*>  10.0.1.0/28      172.16.11.11          0           0 100 i
*>  10.0.1.16/28      172.16.11.11          0           0 100 i
*>  10.0.1.32/28      172.16.11.11          0           0 100 i
*>  10.0.1.48/28      172.16.11.11          0           0 100 i
*>  10.0.1.64/28      172.16.11.11          0           0 100 i
*>  10.0.1.80/28      172.16.11.11          0           0 100 i
```

The R1 router received six BGP prefixes from the ISP1 router. For these prefixes, the next hop IP address is 172.16.11.11.

Step 2

On the R1 router, verify what BGP prefixes are sent to internal BGP peer R2 (10.0.0.33).

```
R1# show ip bgp neighbors 10.0.0.33 advertised-routes
BGP table version is 10, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  10.0.0.0/28      0.0.0.0              0         32768 i
*>  10.0.1.0/28      172.16.11.11          0           0 100 i
*>  10.0.1.16/28      172.16.11.11          0           0 100 i
*>  10.0.1.32/28      172.16.11.11          0           0 100 i
*>  10.0.1.48/28      172.16.11.11          0           0 100 i
*>  10.0.1.64/28      172.16.11.11          0           0 100 i
*>  10.0.1.80/28      172.16.11.11          0           0 100 i

Total number of prefixes 7
```

You can see that all the prefixes that are received from the external BGP peer are sent to the internal BGP peer.

Step 3

On the R2 router, verify if all six prefixes that are originated in AS 100 are received from R1 router.

```
R2# show ip bgp
BGP table version is 16, local router ID is 10.0.0.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>i 10.0.0.0/28      10.0.0.1             0         100   0 i
*>  10.0.0.32/28      0.0.0.0              0         32768 i
*>i 10.0.0.48/28      10.0.0.49             0         100   0 i
* i 10.0.1.0/28      172.16.11.11          0         100   0 100 i
* i 10.0.1.16/28      172.16.11.11          0         100   0 100 i
```

* i 10.0.1.32/28	172.16.11.11	0	100	0 100 i
* i 10.0.1.48/28	172.16.11.11	0	100	0 100 i
* i 10.0.1.64/28	172.16.11.11	0	100	0 100 i
* i 10.0.1.80/28	172.16.11.11	0	100	0 100 i
* i 10.0.3.0/28	172.16.34.34	0	100	0 300 i
* i 10.0.3.16/28	172.16.34.34	0	100	0 300 i
* i 10.0.3.32/28	172.16.34.34	0	100	0 300 i
* i 10.0.3.48/28	172.16.34.34	0	100	0 300 i

On the R2 router, you should see all six prefixes from AS 100 in the BGP routing table.

Step 4

On the R1 router, verify what BGP prefixes are sent to internal BGP peer R3 (10.0.0.17).

```
R1# show ip bgp neighbors 10.0.0.17 advertised-routes
BGP table version is 10, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.0.0/28      0.0.0.0              0         32768 i
*> 10.0.1.0/28      172.16.11.11         0          0 100 i
*> 10.0.1.16/28     172.16.11.11         0          0 100 i
*> 10.0.1.32/28     172.16.11.11         0          0 100 i
*> 10.0.1.48/28     172.16.11.11         0          0 100 i
*> 10.0.1.64/28     172.16.11.11         0          0 100 i
*> 10.0.1.80/28     172.16.11.11         0          0 100 i

Total number of prefixes 7
```

You can see all prefixes received from external BGP peer are sent to internal BGP peer.

Step 5

On the R3 router, verify if all six prefixes that are originated in AS 100 are received from R1 router.

```
R3# show ip bgp
BGP table version is 16, local router ID is 10.0.0.17
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>i 10.0.0.0/28      10.0.0.1              0         100   0 i
*> 10.0.0.16/28      0.0.0.0              0         32768 i
*>i 10.0.0.48/28     10.0.0.49             0         100   0 i
* i 10.0.1.0/28      172.16.11.11         0         100   0 100 i
* i 10.0.1.16/28     172.16.11.11         0         100   0 100 i
* i 10.0.1.32/28     172.16.11.11         0         100   0 100 i
* i 10.0.1.48/28     172.16.11.11         0         100   0 100 i
* i 10.0.1.64/28     172.16.11.11         0         100   0 100 i
* i 10.0.1.80/28     172.16.11.11         0         100   0 100 i
* i 10.0.3.0/28      172.16.34.34         0         100   0 300 i
* i 10.0.3.16/28     172.16.34.34         0         100   0 300 i
* i 10.0.3.32/28     172.16.34.34         0         100   0 300 i
* i 10.0.3.48/28     172.16.34.34         0         100   0 300 i
```

On the R3 router, you should see all six prefixes from AS 100 in the BGP routing table.

Step 6

On the R2 router, verify what BGP prefixes are sent to internal BGP peer R4 (10.0.0.49).

```
R2# show ip bgp neighbors 10.0.0.49 advertised-routes
BGP table version is 16, local router ID is 10.0.0.33
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
```

```

Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*>  10.0.0.32/28    0.0.0.0          0         32768 i

Total number of prefixes 1

```

You can see that no prefixes are received from internal BGP peer and sent to internal BGP peer. There is a single prefix, locally originated, that is sent to internal BGP peer.

Step 7

On the R3 router, verify what BGP prefixes are sent to internal BGP peer R4 (10.0.0.49).

```

R3# show ip bgp neighbors 10.0.0.49 advertised-routes
BGP table version is 16, local router ID is 10.0.0.17
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*>  10.0.0.16/28    0.0.0.0          0         32768 i

Total number of prefixes 1

```

You can see that no prefixes are received from internal BGP peer and sent to internal BGP peer. There is a single prefix, locally originated, that is sent to internal BGP peer.

R4 router has no internal BGP session to the R1 router.

Step 8

On the R4 router, you will not find any prefixes from AS 100.

```

R4# show ip bgp
BGP table version is 8, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

      Network      Next Hop      Metric LocPrf Weight Path
*>i  10.0.0.16/28    10.0.0.17          0      100       0 i
*>i  10.0.0.32/28    10.0.0.33          0      100       0 i
*>   10.0.0.48/28    0.0.0.0          0         32768 i
*>   10.0.3.0/28     172.16.34.34       0           0 300 i
*>   10.0.3.16/28    172.16.34.34       0           0 300 i
*>   10.0.3.32/28    172.16.34.34       0           0 300 i
*>   10.0.3.48/28    172.16.34.34       0           0 300 i

```

Step 9

Configure BGP session between R1 and R4 routers.

BGP session is sourced from Loopback 1 interface. Router R1 Loopback 1 interface has IP address 10.0.0.1 and router R4 Loopback 1 interface has IP address 10.0.0.49. Both routers are in the BGP AS 1.

```

R1(config)# router bgp 1
R1(config-router)# neighbor 10.0.0.49 remote-as 1
R1(config-router)# neighbor 10.0.0.49 update-source Loopback 1

```

```

R4(config)# router bgp 1
R4(config-router)# neighbor 10.0.0.1 remote-as 1
R4(config-router)# neighbor 10.0.0.1 update-source Loopback 1

```

Step 10

On the R1 router, verify what BGP prefixes are sent to internal BGP peer R4 (10.0.0.49).

```
R1# show ip bgp neighbors 10.0.0.49 advertised-routes
BGP table version is 11, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	10.0.0.0/28	0.0.0.0	0		32768	i
*>	10.0.1.0/28	172.16.11.11	0		0	100 i
*>	10.0.1.16/28	172.16.11.11	0		0	100 i
*>	10.0.1.32/28	172.16.11.11	0		0	100 i
*>	10.0.1.48/28	172.16.11.11	0		0	100 i
*>	10.0.1.64/28	172.16.11.11	0		0	100 i
*>	10.0.1.80/28	172.16.11.11	0		0	100 i

Total number of prefixes 7

You can see all prefixes received from external BGP peer are sent to internal BGP peer.

Step 11

On the R4 router, verify if all six prefixes that are originated in AS 100, are received from R1 router.

On the R4 router, you should see all six prefixes from AS 100 in the BGP routing table.

```
R4# show ip bgp
BGP table version is 9, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	10.0.0.0/28	10.0.0.1	0	100	0	i
*>i	10.0.0.16/28	10.0.0.17	0	100	0	i
*>i	10.0.0.32/28	10.0.0.33	0	100	0	i
*>	10.0.0.48/28	0.0.0.0	0		32768	i
* i	10.0.1.0/28	172.16.11.11	0	100	0	100 i
* i	10.0.1.16/28	172.16.11.11	0	100	0	100 i
* i	10.0.1.32/28	172.16.11.11	0	100	0	100 i
* i	10.0.1.48/28	172.16.11.11	0	100	0	100 i
* i	10.0.1.64/28	172.16.11.11	0	100	0	100 i
* i	10.0.1.80/28	172.16.11.11	0	100	0	100 i
*>	10.0.3.0/28	172.16.34.34	0		0	300 i
*>	10.0.3.16/28	172.16.34.34	0		0	300 i
*>	10.0.3.32/28	172.16.34.34	0		0	300 i
*>	10.0.3.48/28	172.16.34.34	0		0	300 i

Now all routers in the AS 1 have same BGP routes.

Step 12

BGP session between R2 and R3 routers is not needed in this lab setup, because R2 and R3 have no external BGP sessions. But in order to be consistent and prepare BGP to accommodate future expansions, configure BGP session between R2 and R3 routers.

BGP session is sourced from Loopback 1 interface. Router R2 Loopback 1 interface has IP address 10.0.0.33 and router R3 Loopback 1 interface has IP address 10.0.0.17. Both routers are in the BGP AS 1.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 10.0.0.17 remote-as 1
R2(config-router)# neighbor 10.0.0.17 update-source Loopback 1
```

```
R3(config)# router bgp 1
R3(config-router)# neighbor 10.0.0.33 remote-as 1
R3(config-router)# neighbor 10.0.0.33 update-source Loopback 1
```

Step 13

On the R4 router, verify what BGP prefixes are sent to external BGP peer ISP3B (172.16.34.34).

```
R4# show ip bgp neighbors 172.16.34.34 advertised-routes
BGP table version is 9, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>i 10.0.0.0/28      10.0.0.1             0      100      0 i
*>i 10.0.0.16/28     10.0.0.17            0      100      0 i
*>i 10.0.0.32/28     10.0.0.33            0      100      0 i
*> 10.0.0.48/28     0.0.0.0              0                   32768 i

Total number of prefixes 4
```

You can see there are no prefixes from AS 100.

Step 14

On the R4 router, verify why routes from AS 100 are not sent to external BGP peer ISP3B (172.16.34.34).

Examine single BGP route from AS 100.

```
R4# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 0
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 2
  100
    172.16.11.11 (inaccessible) from 10.0.0.1 (10.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal
```

The output shows that route 10.0.1.0/28 has inaccessible next hop 172.16.11.11. This inaccessibility is a reason that the route is not marked as a good route and is not advertised toward external BGP peer.

```
R4# show ip route 172.16.11.11
% Subnet not in table
```

On the R4 router, there is no route to the next hop 172.16.11.11 in the IP routing table.

Step 15

On the R1 router, verify what is next hop IP address of the BGP route 10.0.1.0/28.

```
R1# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 3
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  100
    172.16.11.11 from 172.16.11.11 (10.0.1.81)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Output shows that route 10.0.1.0/28 has valid next hop 172.16.11.11.

Transit Network Using External Next Hops

All BGP-speaking routers within the AS get information about external networks with the next-hop attribute. The next-hop attribute is set to the far end of the EBGP sessions reaching the border routers of the AS.

- All EBGP peers must be reachable by all BGP-speaking routers within the AS.

- EBGp next hops shall be announced using the IGP.
 - Redistribute connected interfaces into the IGP at the edge routers.
 - Include links to EBGp neighbors into the IGP and configure them as passive interfaces.

Routers use a recursive routing mechanism when they determine how to forward IP packets toward external destinations. When BGP routes are used in the routing table, the router checks how it would have reached the next-hop address, and it installs the BGP route with the same forwarding indication as for the route that is used to reach the next-hop IP address.

To get the recursive routing to work, the router must resolve all possible next-hop references that use information in the routing table, which is already there. The IGP that is used within the AS must carry this information.

One way of making the IGP carry the information that is necessary to resolve the BGP next-hop addresses is to make sure that all the border routers, which contain the EBGp sessions, redistribute connected subnets into the IGP using the **redistribute connected** routing protocol configuration command. Because EBGp sessions are established between routers using a directly connected interface, the far end of the EBGp sessions is an IP address within the directly connected subnet. By redistributing the connected interfaces into the IGP, the border routers allow next-hop references to be resolvable by all routers within the AS.

External subnets that are redistributed into the IGP might appear as external IGP routes, depending on what IGP is configured within the AS. There are several scalability issues that are associated with external routes in some routing protocols. For example, [OSPF](#) carries each external subnet in a separate [LSA](#) object. If route redistribution is not desirable for any reason, an alternative method is to include the subnet on which the EBGp session is running in the IGP configuration using the **network** command. To prevent the border router from exchanging IGP routing with the border router of the other AS, you must configure the interface as a passive interface. Failure to do so could cause the two autonomous systems to exchange routes using the IGP. In that case, all benefits of having separate autonomous systems would be lost.

Step 16

On the R1 router, redistribute subnet from the [Ethernet](#) 0/0, facing to the ISP1 router, into OSPF routing protocol.

Between routers in the AS 1 (R1, R2, R3, and R4) there is OSPF with process ID 1 in the area 0 preconfigured.

```
R1(config)# route-map toISP1 permit 10
R1(config-route-map)# match interface Ethernet 0/0
R1(config-route-map)# exit
R1(config)# router ospf 1
R1(config-router)# redistribute connected subnets route-map toISP1
```

You need to use **subnets** keyword, because subnet of the network 172.16.0.0/16 is used between R1 and ISP1 routers.

Step 17

On the R4 router, examine BGP route 10.0.1.0/28 from AS 100 again.

```
R4# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 10
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 3
  100
    172.16.11.11 (metric 20) from 10.0.0.1 (10.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Output shows that route 10.0.1.0/28 has valid next hop 172.16.11.11 and is selected as best route.

```

R4# show ip route 172.16.11.11
Routing entry for 172.16.11.0/24
  Known via "ospf 1", distance 110, metric 20, type extern 2, forward metric 20
  Last update from 192.168.34.3 on Ethernet0/2, 00:10:23 ago
  Routing Descriptor Blocks:
    192.168.34.3, from 10.0.0.1, 00:10:23 ago, via Ethernet0/2
      Route metric is 20, traffic share count is 1
    * 192.168.24.2, from 10.0.0.1, 00:10:23 ago, via Ethernet0/3
      Route metric is 20, traffic share count is 1

```

On the R4 router, there is also routing information to reach the next hop 172.16.11.11.

Step 18

On the R4 router, verify what BGP prefixes are sent to external BGP peer ISP3B (172.16.34.34).

```

R4# show ip bgp neighbors 172.16.34.34 advertised-routes
BGP table version is 15, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*>i 10.0.0.0/28      10.0.0.1             0      100      0 i
*>i 10.0.0.16/28     10.0.0.17            0      100      0 i
*>i 10.0.0.32/28     10.0.0.33            0      100      0 i
*> 10.0.0.48/28     0.0.0.0              0           32768 i
*>i 10.0.1.0/28      172.16.11.11         0      100      0 100 i
*>i 10.0.1.16/28     172.16.11.11         0      100      0 100 i
*>i 10.0.1.32/28     172.16.11.11         0      100      0 100 i
*>i 10.0.1.48/28     172.16.11.11         0      100      0 100 i
*>i 10.0.1.64/28     172.16.11.11         0      100      0 100 i
*>i 10.0.1.80/28     172.16.11.11         0      100      0 100 i

Total number of prefixes 10

```

You can see there are also prefixes from AS 100 advertised to external BGP peer ISP3B.

Step 19

On the ISP3B router, verify what is next hop IP address of the BGP route 10.0.1.0/28.

```

ISP3B# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 10
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1 100
    172.16.34.4 from 172.16.34.4 (10.0.0.49)
      Origin IGP, localpref 100, valid, external, best

```

Output shows that route 10.0.1.0/28 has different next hop IP address (172.16.34.4), as seen on the R4 router (172.16.11.11). When BGP route is sent to external BGP peer, the next-hop IP address changes.

Transit Network Using Edge Routers as Next Hops

An IBGP peer usually does not modify the next-hop attribute when the BGP update is propagated across IBGP sessions. However, you could configure the BGP router to have a different behavior and set its IP address as the next-hop address even when the BGP updates are sent across IBGP sessions (emulating behavior on EBGp sessions). If you do configure an IBGP router to emulate the behavior of EBGp sessions on the IBGP sessions of the border routers, the BGP updates that are received on the EBGp sessions will be forwarded on the IBGP sessions and the next-hop attribute will be set to the IP address that is used on the local side of the IBGP session. The original next hop, set by the far end of the EBGp session, will be lost.

- **Alternate design:** Next-hop processing is modified at the edge routers.

- Edge routers announce themselves as the next hop in IBGP updates.
- No redistribution of external subnets is necessary.
- This design might result in suboptimal routing if multiple paths to a neighboring AS exist.
- Use default next-hop processing if at all possible.

The receiver of the IBGP information will do recursive routing in the normal way. But the next-hop address that is used will be the IP address of the far end of the IBGP session, because the border router has changed it. The IP address of the far-end IBGP peer is always known in the routing table; otherwise, the IBGP session would not have been established. There is no need for the receiver of the IBGP information to have knowledge of how to reach the far end of the EBGP session, because that IP address is no longer set as the next hop.

To configure the router as the next hop for a BGP-speaking neighbor, use the **neighbor next-hop-self** router configuration command.

Step 20

On the R1 router, remove the previously configured redistribution.

```
R1(config)# router ospf 1
R1(config-router)# no redistribute connected
```

On the R4 router, the BGP route (10.0.1.0/28) next hop IP address 172.16.11.11 will again become inaccessible.

```
R4# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 16
Paths: (1 available, no best path)
  Not advertised to any peer
  Refresh Epoch 3
  100
    172.16.11.11 (inaccessible) from 10.0.0.1 (10.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal
```

Step 21

Configure R1 router to be next hop for all IBGP speaking routers.

The IBGP speaking routers of the R1 router are R2 (10.0.0.33), R3 (10.0.0.17), and R4 (10.0.0.49).

```
R1(config)# router bgp 1
R1(config-router)# neighbor 10.0.0.17 next-hop-self
R1(config-router)# neighbor 10.0.0.33 next-hop-self
R1(config-router)# neighbor 10.0.0.49 next-hop-self
```

Step 22

On the R4 router, examine BGP route 10.0.1.0/28 from AS 100 again.

```
R4# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 22
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 3
  100
    10.0.0.1 (metric 21) from 10.0.0.1 (10.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

Output shows that route 10.0.1.0/28 has valid next hop 10.0.0.1 and is selected as best route. Next hop 10.0.0.1 is R1 Loopback 1 interface.

```
R4# show ip route 10.0.0.1
Routing entry for 10.0.0.1/32
  Known via "ospf 1", distance 110, metric 21, type intra area
  Last update from 192.168.34.3 on Ethernet0/2, 00:46:42 ago
```

Routing Descriptor Blocks:

```
192.168.34.3, from 10.0.0.1, 00:46:42 ago, via Ethernet0/2
Route metric is 21, traffic share count is 1
* 192.168.24.2, from 10.0.0.1, 00:46:42 ago, via Ethernet0/3
Route metric is 21, traffic share count is 1
```

On the R4 router, there is routing information to reach the next hop 10.0.0.1.

Step 23

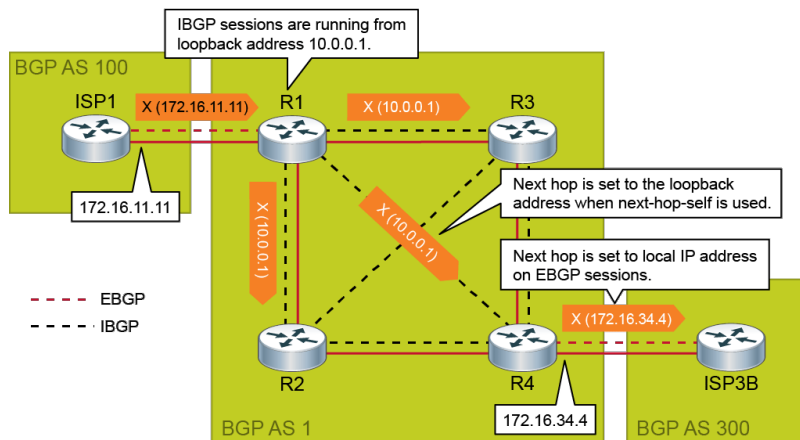
On the ISP3B router, verify what is next hop IP address of the BGP route 10.0.1.0/28.

```
ISP3B# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 22
Paths: (1 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  1 100
    172.16.34.4 from 172.16.34.4 (10.0.0.49)
      Origin IGP, localpref 100, valid, external, best
```

The output shows that route 10.0.1.0/28 has different next hop IP address (172.16.34.4), as seen on the R4 router (10.0.0.1).

Transit Network Using Edge Routers as Next Hops Example

The next-hop attribute is normally not changed on [IBGP](#) updates. When the border router forwards the incoming [EBGP](#) update over an outgoing IBGP session, the border router changes the next-hop address to the [IP address](#) that is used as the source address of the IBGP session.



The receiver of IBGP updates will see next-hop information that indicates a destination, which might not be directly connected. To resolve this problem, it will check its routing table and see if and how the next-hop address can be reached. Then it will route [IP](#) packets with destination addresses that match the network in the [BGP](#) update in the same direction as it would have routed an IP packet with the destination address equal to the IP address in the next-hop attribute. In this case, it is obvious that the next-hop address can be reached, because the IBGP session would not have been established otherwise.

In the figure, ISP1 sends a BGP update about network X. Because it is sending a BGP update over an EBGP session to R1, the next-hop attribute is set to the IP address that is used at the ISP1 side of the EBGP session, 172.16.11.11. R1 can use this information and route packets to network X by forwarding them to ISP1.

R1 also forwards the BGP update on all its IBGP sessions. It changes the next-hop attribute to the IP address of its own loopback interface, so R2, R3, and R4 will get information that they can reach network X by forwarding packets to 10.0.0.1. But that address is not directly connected. The routers will inspect the routing table to see if and how they can reach 10.0.0.1. They can then route packets to network X in the same direction that they would use to route packets to 10.0.0.1.

R4 also forwards the BGP update about network X to ISP3B. This session is an EBGP session, which means that R4 will set the next-hop attribute to its own IP address that is used on that EBGP session, 172.16.34.4.

Differences Between EBGP and IBGP Sessions

Both [EBGP](#) and [IBGP](#) sessions forward [BGP](#) updates; however, they do it in slightly different ways.

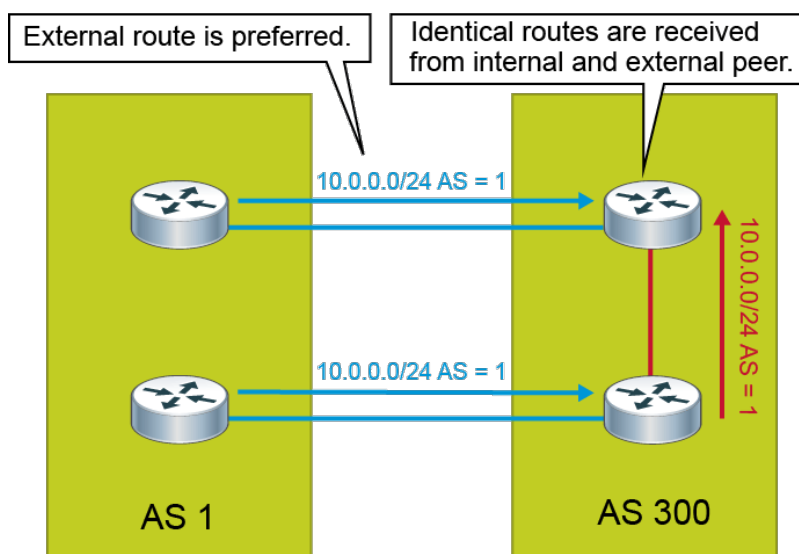
- No BGP attributes are changed in IBGP updates.
- Because of BGP split horizon, routes that are learned from an IBGP peer are not advertised to other IBGP peers.
- Local preference is propagated only over IBGP sessions.
- EBGP peers are directly connected; IBGP peers are usually distant.
- Route selection rules slightly prefer EBGP routes.

Here are the differences between EBGP and IBGP sessions in forwarding BGP updates:

- The router does not change BGP attributes when an update is sent across an IBGP session, unless **next-hop-self** is configured. When a BGP-speaking router sends an update across an EBGP session, the next-hop attribute is always set and the [AS](#) number of the router is prepended to the AS-path attribute.
- IBGP uses split horizon to prevent routing information loops. EBGP does not use split horizon and instead uses the AS path to detect loops. In both cases, a router forwards only the best route and never sends a route back on the session from which it was received. But IBGP split-horizon rules also prohibit a router from forwarding any information that is received on an IBGP session to another IBGP session.
- IBGP border routers remove the local preference attribute from a BGP route before the BGP update is sent over an EBGP session. This difference means that the local preference attribute is distributed on IBGP sessions only.
- Two routers with an EBGP session between them normally establish the session using the [IP addresses](#) from a common, shared subnet. Using the shared subnet to establish the session guarantees that the two routers can exchange [IP](#) packets without any [IGP](#) running between them. Also, recursive routing will always succeed because the next-hop address is reachable using a directly connected route.
- IBGP sessions are normally established between all routers in the AS in a full mesh. But all routers in an AS might not have physical connections to every other router within the AS. Because IBGP sessions are established between routers using IP addresses of different subnets, an IGP must be running within the AS in order to establish IBGP sessions.
- BGP route selection rules slightly favor EBGP routes over equivalent IBGP routes.

Differences Between EBGP and IBGP Sessions Example

This example illustrates the preference of the EBGP route.



- Whenever identical routes are received from IBGP and EBGP peers, the route from the EBGP peer is preferred.

One of the default goals of transit packet forwarding is to propagate the transit packet toward the downstream AS as soon as possible. A border router that receives otherwise equivalent routes to the same destination over both an EBGp session and an IBGP session will prefer the information that is received through the EBGp session.



Equivalent routes are routes that have equal BGP path attributes used in the BGP route selection rules (weight, local preference, AS-path length, origin, [MED](#)).

In the figure, the upper router in AS 300 receives BGP updates about network 10.0.0.0/24 over two different paths. One update is received over the EBGp session to AS 1. The other update is received over the IBGP session to the lower router in AS 300. All essential attributes are the same, so route selection cannot be made easily.

The upper router in AS 300 realizes that IP packets with destination addresses within network 10.0.0.0/24 should sooner rather than later leave AS 300. It is better to make them leave the AS right away. So the update that was received on the EBGp session is preferred over the update that was received on the IBGP session.

Scalability Limitations of IBGP-Based Transit Backbones

The requirements that are associated with [IBGP](#)-based transit backbones can result in limited scalability.

Transit backbone requires IBGP full mesh between all core routers.

- Large number of TCP sessions
- Unnecessary, duplicate routing traffic

There are two scalability solutions:

- Route reflectors
- BGP confederations

IBGP split-horizon rules mandate an IBGP connection between every border router and every other [BGP](#) router in an [AS](#).

The general design rule in IBGP design is to have a full mesh of IBGP sessions. But, a full mesh of IBGP sessions among n number of routers would require $(n * (n - 1)) / 2$ IBGP sessions. For example, a full mesh between 10 routers would require $(10 * 9) / 2 = 45$ IBGP sessions.

Because every IBGP session on a router uses a separate [TCP](#) session, an update that must be sent by the router to all IBGP peers must be sent on each of the TCP sessions. If a router is attached to the rest of the network over just a single link, this single link has to carry all [TCP/IP](#) packets for all IBGP sessions. This situation results in duplication of the update over the single link.

Two solutions are available:

- The route reflector solution modifies the IBGP split-horizon rules and allows a particular router to forward (under certain conditions) incoming IBGP updates to a select group of IBGP neighbors. The router performing this function is the "route reflector."
- The BGP confederations solution introduces the concept of a number of smaller autonomous systems within the original AS. These smaller autonomous systems exchange BGP updates among themselves using intraconfederation [EBGP](#) sessions.

Summary

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

- All BGP routing updates carry the mandatory well-known attribute AS-path. The AS-path attribute is not changed when the BGP prefix is propagated across IBGP sessions.
- The IBGP multipath load sharing feature enables the BGP speaking router to select multiple IBGP paths as the best paths to a destination.
- IBGP split horizon prevents routing information loops within the AS.
- All BGP routers within an AS must have IBGP sessions with every other BGP router in the AS.
- For recursive routing to work, a router must resolve all possible next-hop references that use information in the routing table.
- You can configure an edge router to set its IP address as the next-hop address even when the BGP updates are sent across IBGP sessions.
- BGP attributes are not changed when an update is sent across an IBGP session unless next-hop-self is configured.
- The full-mesh IBGP requirement in the transit AS creates scalability issues in the number of TCP sessions and unnecessary, duplicate routing traffic. IBGP scalability solutions to these issues exist.

Forwarding Packets in Transit AS

Overview

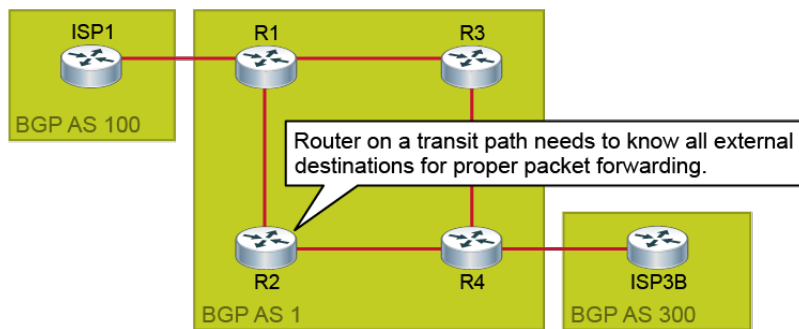
A transit [AS](#) requires interaction between [EBGP](#) and [IBGP](#) and between IBGP and an [IGP](#) in the transit AS. This lesson describes packet forwarding through a transit AS and discusses the requirements for successful packet forwarding, such as recursive route lookup and an IGP in the transit AS. This lesson concludes with a discussion of the interaction between IBGP and an IGP running within the transit AS.

Upon completing this lesson, you will be able to:

- Describe packet forwarding in a transit AS
- Explain how recursive lookup functions in Cisco IOS software
- Explain the need for an IGP in a transit backbone that is running BGP on all routers
- Describe interactions between BGP and IGP in a transit AS
- Describe how to change BGP administrative distances
- Explain the potential problems that might arise from BGP and IGP interaction

Packet Forwarding in Transit AS

When BGP updates have propagated through the transit AS to all neighboring autonomous systems, the IP traffic can start to flow.



All core routers need external routers for proper packet forwarding.

- Redistributing can overload IGP resources.
- IBGP is preferred for scalability.

In the figure, ISP3B router forwards to R4 IP packets with the destination address matching a network in AS 100. R4 checks its routing table and finds that there is a BGP route for that destination. The BGP route has a next-hop reference, which points to the far end of the EBGP session between ISP1 and R1. So R4 once again checks the routing table and finds that it should forward the packet to R2 in this case.

Thus, R2 receives the IP packet with a destination address indicating a host within AS 100. To be able to forward this packet, R2 must have a matching route in its routing table. A default route or gateway of last resort is not appropriate because in the next instant R2 could receive another packet, coming from the other direction and destined for AS 300.

The conclusion is that both R2 and R3, to handle all possible cases, must have routing information to all the external networks that R1 and R4 have. The only scalable way of providing routers with this information is to update R2 and R3 with IBGP from both R1 and R4.

In theory, the external information that R1 and R4 receive could be redistributed by these ingress routers into the IGP in use within the transit AS. However, no IGP can handle the volume of information that BGP can. So there would always be a risk that the IGP would break because of information overload, causing a total network meltdown in the AS. The volume of routing information that the BGP carries in the contemporary Internet long ago passed the limits of what it is possible to carry in any IGP.

- Routes that are learned via BGP do not have an outgoing interface associated with them in the routing table.
- Recursive lookup is performed to forward IP packets toward external destinations.

```
R2# show ip route 10.0.1.0 255.255.255.240
Routing entry for 10.0.1.0/28
  Known via "bgp 1", distance 200, metric 0
  Tag 100, type internal
  Last update from 10.0.0.1 16:37:59 ago
  Routing Descriptor Blocks:
  * 10.0.0.1, from 10.0.0.1, 16:37:59 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
    Route tag 100
    MPLS label: none
```

- Observe that the route that is known via BGP has no outgoing interface, but it has a BGP next hop.

```
R2# show ip route 10.0.0.1
Routing entry for 10.0.0.1/32
  Known via "ospf 1", distance 110, metric 11, type intra area
  Last update from 192.168.12.1 on Ethernet0/2, 17:23:10 ago
```

Routing Descriptor Blocks:

```
* 192.168.12.1, from 10.0.0.1, 17:23:10 ago, via Ethernet0/2
  Route metric is 11, traffic share count is 1
```

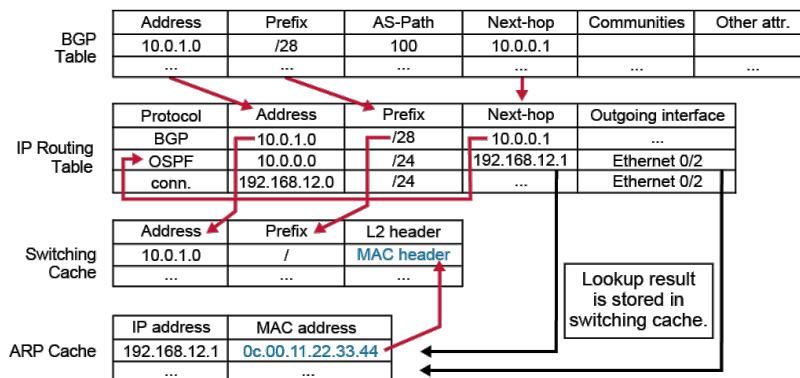
- The route toward BGP next hop is known via OSPF and has an outgoing interface.

A BGP route is installed in the IP routing table of a router only if the IP address in the next-hop attribute is reachable according to the information already in the routing table. The installed BGP route contains a reference to that next-hop address. So, the network will be reachable via an [IP address](#), which may or may not be directly connected. Because there is no clear reference to a physical interface, the BGP route is installed in the IP routing table without any information about outgoing interface.

The router must evaluate the recursive reference to the BGP next hop sooner or later in order to allow packet forwarding toward external destinations. The point in time when the recursive reference is resolved depends on the IP switching mechanism that the router uses. At the latest, the router performs the recursive route lookup when an IP packet with a destination address that matches the BGP route should be forwarded. The router determines which outgoing interface should be used and which Layer 2 address to assign (if applicable). The router creates a cache entry so that successive IP packets to the same destination can be routed using the same outgoing interface and Layer 2 address.

Recursive Lookup in Cisco IOS Software

Recursive lookup refers to the [BGP](#) routes for which another routing lookup is required in order to resolve the BGP route next hop.



- Entries in routing table are built from BGP table.
- Outgoing interface is never associated with a BGP route.
- ARP cache lookup is performed to build Layer 2 header.

The figure presents the steps in the recursive lookup process in Cisco IOS software. The router has received a BGP update about network 10.0.1.0/28. It was associated with an [AS-path](#) attribute set to 100, a next-hop attribute set to the [IP address](#) 10.0.0.1. Some other attributes were also carried with the update.

Because the next-hop address 10.0.0.1 is reachable according to the routing table, the BGP route is also installed in the routing table. Network number, subnet mask, and next-hop attributes are inherited from the BGP table. No outgoing interface is assigned.

When an [IP](#) packet with a destination in network 10.0.1.0/28 is received, the router searches the routing table and finds the installed BGP route. The router takes the indicated next-hop address 10.0.0.1 and searches the routing table again. It now finds a match with the [OSPF](#) route to subnet 10.0.0.0/24. The 10.0.0.0/24 route has an outgoing interface set to interface [Ethernet](#) 0/2 and a next hop set to 192.168.12.1. So, the packets that are destined for network 10.0.1.0/28 should be forwarded via 192.168.12.1, which is directly reachable over Ethernet 0/2. The [ARP](#) table is used to find the [MAC address](#) for IP address 192.168.12.1. The MAC address is used to forward the IP packet to network 10.0.1.0/28 out the Ethernet 0/2 interface. The [MAC](#) header is stored in the cache for successive packets to network 10.0.1.0/28.



The example illustrates the recursive lookup performed when the router uses cache-based IP switching mechanisms (for example, fast switching or optimum switching).

- Traditional Cisco IOS software switching mechanisms perform recursive lookup when forwarding the first packet.
 - Fast switching, optimum switching.
- CEF precomputes the routing table.
 - All recursive lookups are performed while the routing table is built.

Traditional Cisco IOS switching mechanisms used the traffic-driven, cache-based switching approach. Both fast switching and optimum switching populate the IP switching cache on demand, meaning that before any IP packets are forwarded, the cache is empty. After the first packet to a specific destination arrives, all routing table lookups are done, including recursive lookup in the case of a BGP route. The result of the lookup is cached for later use when successive packets for the same destination arrive. The process is repeated for every specific destination.

Cisco Express Forwarding (CEF) prebuilds a complete IP forwarding table, called the [FIB](#), that is based on the IP routing table. After the router installs a routing entry into its routing table, incoming routing information updates trigger the recursive lookup. With the recursive lookup, the outgoing interface and the actual physical next hop of the route are determined. MAC address resolution and MAC header

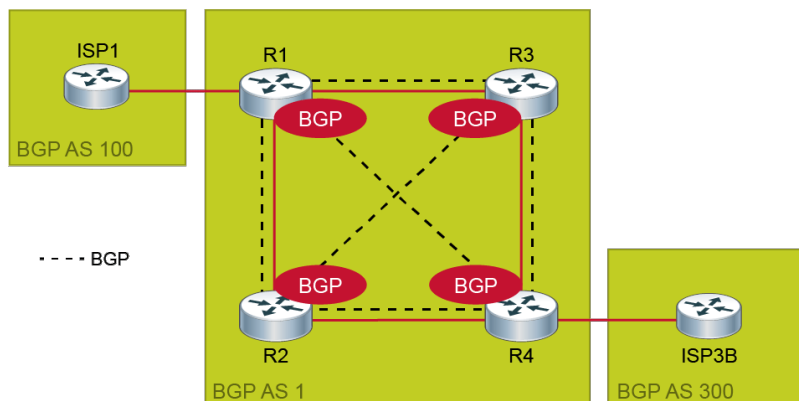
generation are still traffic-driven and stored in the cache.

Routing Protocols in Transit AS

Some network designers base their network design on the wrong assumption that an internal routing protocol is not needed in a transit AS where all routers run BGP.

However, the internal routing protocol is still needed inside an AS for two reasons:

- To provide routing information that is needed to establish the IBGP sessions.
- To resolve next-hop references (recursive routing).



With BGP running on all core routers, is an IGP still needed in the core? An IGP is needed to resolve BGP next hops and perform fast convergence after a failure in the core network.

When R4 in the figure receives an IP packet with the destination in AS 100, it does a recursive lookup to find the outgoing interface to be used for packet forwarding. It performs the recursive lookup that is based on the IGP information. If there is suddenly an internal problem within AS 1, and the next-hop address is reachable a different way, the IGP determines this fact. The router changes the IGP route to the next-hop network because of the newly received IGP route information, and all cache entries that rely on the old information are invalidated. The next recursive lookup that R4 performs will indicate a different outgoing interface than before the problem occurred.

During the IGP convergence process, the BGP routing is not affected. The only routing updates that are exchanged during the transit AS convergence are IGP updates describing how to reach internal destinations (including the far ends of the EBGP sessions).

The packet forwarding to external destinations thus benefits from the high-speed convergence that the IGP offers. The faster the IGP determines that it should use an alternate path within the AS to reach the next-hop address, the faster it will re-establish IP connectivity toward external destinations.

The conclusion is that an IGP is still needed inside a transit AS, and the network will work better if it is an IGP with fast convergence.

- Core routers need to run BGP and an IGP.
- BGP carries all external routes.
- The IGP propagates BGP next hops and other core subnets only.
- All customer routes are also carried in BGP.
 - Reduces IGP topology database.
 - Removes customer-caused route flaps from IGP; IGP becomes more stable.

Both BGP and the configured IGP should be configured on all core routers inside the transit AS. The IGP should carry as little information as possible. Ideally, this information includes only the links within the core network, the loopback interfaces, and the external subnets that are used in EBGP sessions with the neighboring autonomous systems. This information is enough to establish IBGP sessions and resolve next-hop addresses. The IGP will also work better if it carries less routing information.

No routes external to the transit AS should ever be redistributed by any router from BGP into the IGP. All external routes should be in BGP only.

In autonomous systems that provide customer connectivity (not only transit service), it is also highly recommended that the customer networks be carried in BGP. This

approach reduces the amount of information in the IGP and increases IGP stability.

BGP and IGP Interaction

Ideally, [BGP](#) and the [IGP](#) carry two different sets of routing information. BGP carries the routes that are received from other autonomous systems and the routes that belong to the local [AS](#) and should be announced to other autonomous systems. The IGP carries only enough information to establish [IBGP](#) sessions and resolve the [EBGP](#) next-hop addresses via the IGP routing tables.

Ideally, there will be no interaction between BGP and the IGP.

- BGP carries external and customer routes.
- The IGP carries only core subnets.
- External route flaps do not affect IGP.
- Failures internal to the network do not affect BGP as long as the BGP next hop remains reachable.
- The only link between BGP and the IGP should be the recursive lookup.

The IGP will provide reachability toward the BGP next-hop addresses only if it is not disturbed by external updates from other autonomous systems.

BGP should take care of the external information. As long as the IGP finds a usable way to the BGP next hops, the BGP does not need to do any recalculation because of internal problems within the AS.

Sometimes, BGP and the IGP will propagate the same route.

- Usually stems from bad network design.
- In this case, routes are determined in EBGP/IGP/IBGP order based on administrative distances of the routes.

Routing Protocol Default Administrative Distance

Routing Protocol	Default Administrative Distance
EBGP	20
IGP	90 — 170
IBGP	200

Sometimes the interaction between BGP and the IGP is not ideal, for a number of reasons, including bad network design. In the worst case, the same networks might be carried in both the IGP and BGP. For example, the subnets connecting the AS with neighboring autonomous systems have to be announced via the IGP to enable next-hop resolution. These subnets may also be announced via BGP by the remote AS or the local AS. In any case, information about the same IP prefix will appear in both the IGP and the BGP data structures.

When the router installs routing information into the routing table, it checks to see whether there are several sources of information for a particular [IP](#) prefix. If so, the router installs the information that it determines is most reliable. The [AD](#) determines which source to use.

BGP considers both EBGP and IBGP routes in the BGP selection process. BGP will therefore never try to install both an EBGP route and an IBGP route for the same destination. Comparison between ADs will thus occur only when two different protocols carry the same destination network.

If BGP selects an EBGP route as the best route for a given destination network, it will try to install that route with a very low AD. So, the routes that are learned via EBGP have a high likelihood of being installed in the routing table.

If BGP selects an IBGP route as the best, it will try to install it with a high AD. Thus, the routes that are learned via IBGP have a low likelihood of being installed in the routing table.

All IGP, such as [EIGRP](#), [OSPF](#), [IS-IS](#), and so on, have a medium likelihood of being installed. The ADs for IGP fall between the ADs of EBGP and IBGP.

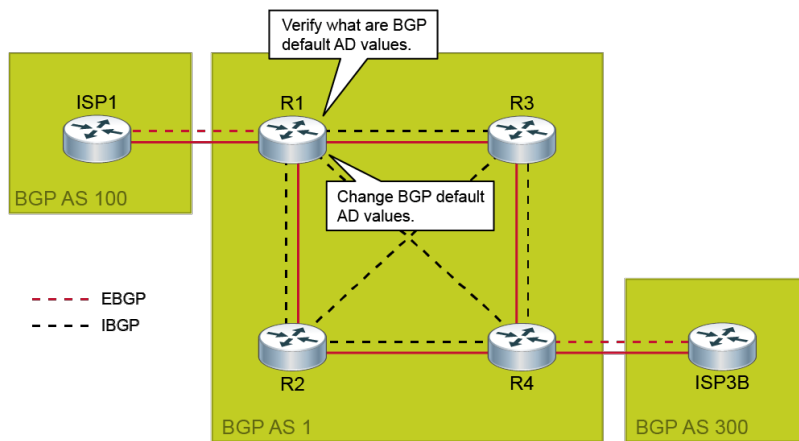


The reason for giving EBGP a low default AD is because EBGP indicates routes external to the local AS. IP packets with destination addresses to those networks should leave the AS sooner rather than later. It is, in most cases, better that they leave the AS right away.

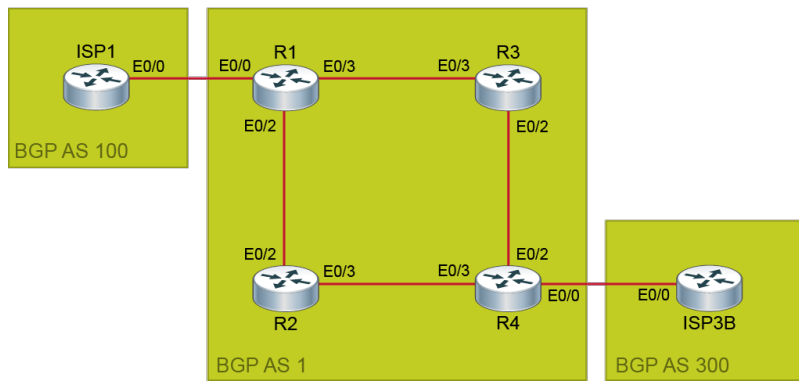
Discovery 6: BGP Administrative Distance

Overview

Through this discovery, you will learn how to change BGP administrative distances.



Topology



Job Aids

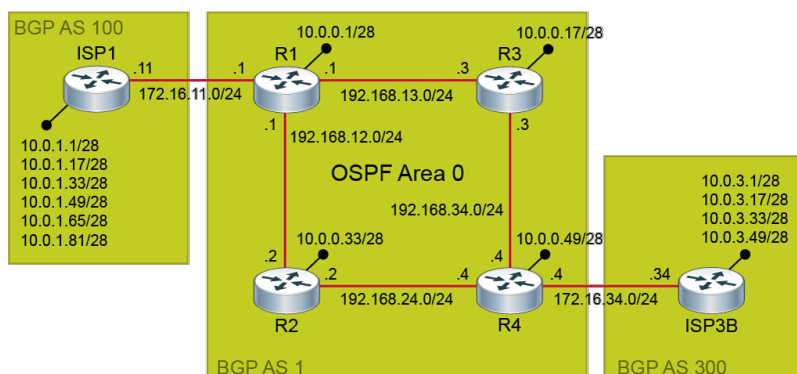


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
ISP1	Ethernet 0/0	172.16.11.1/24	Connection to R1
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R1	Ethernet 0/3	192.168.13.1/24	Connection to R3
R1	Loopback 1	10.0.0.1/28	Loopback simulates LAN network
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Ethernet 0/3	192.168.24.2/24	Connection to R4
R2	Loopback 1	10.0.0.33/28	Loopback simulates LAN network
R3	Ethernet 0/2	192.168.34.3/24	Connection to R4
R3	Ethernet 0/3	192.168.13.3/24	Connection to R1
R3	Loopback 1	10.0.0.17/28	Loopback simulates LAN network
R4	Ethernet 0/0	172.16.34.4/24	Connection to ISP3B
R4	Ethernet 0/2	192.168.34.4/24	Connection to R3
R4	Ethernet 0/3	192.168.24.4/24	Connection to R2
R4	Loopback 1	10.0.0.49/28	Loopback simulates LAN network
ISP3B	Ethernet 0/0	172.16.34.34/24	Connection to R4
ISP3B	Loopback 1 Loopback 2 Loopback 3 Loopback 4	10.0.3.1/28 10.0.3.17/28 10.0.3.33/28 10.0.3.49/28	Loopbacks simulate LAN networks

IP addresses and IGP are preconfigured as shown in the topology below:



BGP is also configured as EBGp (ISP1 to R1 and ISP3B to R4) and as IBGP (full mesh between R1, R2, R3, and R4).

BGP Administrative Distance

Discovery Steps

Step 1

On the R1 router, verify what are BGP administrative distances.

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "bgp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Neighbor(s):
    Address           FiltIn FiltOut DistIn DistOut Weight RouteMap
    10.0.0.17
    10.0.0.33
    10.0.0.49
    172.16.11.11
  Maximum path: 1
  Routing Information Sources:
    Gateway           Distance    Last Update
    10.0.0.17         200        00:04:14
    10.0.0.33         200        00:04:33
    10.0.0.49         200        00:04:09
    172.16.11.11     20         00:04:33
  Distance: external 20 internal 200 local 200

Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 10.0.0.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.0.0 0.0.0.255 area 0
    192.168.0.0 0.0.255.255 area 0
  Routing Information Sources:
    Gateway           Distance    Last Update
    10.0.0.17         110        00:09:56
    10.0.0.33         110        00:10:11
    10.0.0.49         110        00:09:32
  Distance: (default is 110)
```

The BGP external AD is 20, internal and local AD are 200. You can also learn OSPF AD value.

Changing Administrative Distance of BGP Routes

You can change BGP administrative distances, to allow the use of BGP route, that could be a better route to a node.

```
router(config-router)# distance bgp external-distance internal-
distance local-distance
```

- This command sets the AD for EBGp, IBGP, and local routes.
- This change applies only to routes received after the command has been entered (similar to filters).
- Defaults: EBGp routes have a distance of 20; IBGP and local routes have a distance of 200.
- *The defaults are usually correct; do not change them.*

Step 2

On the R1 router, change EBGp AD into 100 and IBGP AD into 190.
Leave default BGP local AD.

```
R1(config)# router bgp 1
R1(config-router)# distance bgp 100 190 200
```

Step 3

On the R1 router, verify what are BGP administrative distances again.

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "bgp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  IGP synchronization is disabled
  Automatic route summarization is disabled
  Neighbor(s):
    Address          FiltIn FiltOut DistIn DistOut Weight RouteMap
    10.0.0.17
    10.0.0.33
    10.0.0.49
    172.16.11.11
  Maximum path: 1
  Routing Information Sources:
    Gateway          Distance      Last Update
    10.0.0.17         200          00:19:41
    10.0.0.33         200          00:20:00
    10.0.0.49         200          00:19:35
    172.16.11.11      20           00:19:59
    Distance: external 100 internal 190 local 200
<... output omitted ...>
```

The BGP EBGP and IBGP ADs changed.

Step 4

On the R1 router, verify IP routing table.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-
user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 18 subnets, 2 masks
C       10.0.0.0/28 is directly connected, Loopback1
L       10.0.0.1/32 is directly connected, Loopback1
B       10.0.0.16/28 [200/0] via 10.0.0.17, 01:21:20
O       10.0.0.17/32 [110/11] via 192.168.13.3, 01:22:35, Ethernet0/3
B       10.0.0.32/28 [200/0] via 10.0.0.33, 01:21:39
O       10.0.0.33/32 [110/11] via 192.168.12.2, 01:22:50, Ethernet0/2
B       10.0.0.48/28 [200/0] via 10.0.0.49, 01:21:14
O       10.0.0.49/32 [110/21] via 192.168.13.3, 01:22:11, Ethernet0/3
        [110/21] via 192.168.12.2, 01:22:11, Ethernet0/2
B       10.0.1.0/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.1.16/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.1.32/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.1.48/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.1.64/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.1.80/28 [20/0] via 172.16.11.11, 01:21:39
B       10.0.3.0/28 [200/0] via 10.0.0.49, 01:21:14
B       10.0.3.16/28 [200/0] via 10.0.0.49, 01:21:14
B       10.0.3.32/28 [200/0] via 10.0.0.49, 01:21:14
B       10.0.3.48/28 [200/0] via 10.0.0.49, 01:21:14
    172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.11.0/24 is directly connected, Ethernet0/0
L       172.16.11.1/32 is directly connected, Ethernet0/0
    192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/24 is directly connected, Ethernet0/2
L       192.168.12.1/32 is directly connected, Ethernet0/2
    192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.13.0/24 is directly connected, Ethernet0/3
L       192.168.13.1/32 is directly connected, Ethernet0/3
O       192.168.24.0/24 [110/20] via 192.168.12.2, 01:22:21, Ethernet0/2
O       192.168.34.0/24 [110/20] via 192.168.13.3, 01:22:21, Ethernet0/3
```

You can still see BGP routes are using default AD values.

Step 5

On the R1 router, clear BGP sessions.

```
R1# clear ip bgp *
R1#
%BGP-5-ADJCHANGE: neighbor 10.0.0.17 Down User reset
%BGP_SESSION-5-
ADJCHANGE: neighbor 10.0.0.17 IPv4 Unicast topology base removed from session User reset
%BGP-5-ADJCHANGE: neighbor 10.0.0.33 Down User reset
%BGP_SESSION-5-
ADJCHANGE: neighbor 10.0.0.33 IPv4 Unicast topology base removed from session User reset
%BGP-5-ADJCHANGE: neighbor 10.0.0.49 Down User reset
%BGP_SESSION-5-
ADJCHANGE: neighbor 10.0.0.49 IPv4 Unicast topology base removed from session User reset
%BGP-5-ADJCHANGE: neighbor 172.16.11.11 Down User reset
%BGP_SESSION-5-
ADJCHANGE: neighbor 172.16.11.11 IPv4 Unicast topology base removed from session User reset
%BGP-5-ADJCHANGE: neighbor 10.0.0.33 Up
%BGP-5-ADJCHANGE: neighbor 10.0.0.17 Up
%BGP-5-ADJCHANGE: neighbor 172.16.11.11 Up
%BGP-5-ADJCHANGE: neighbor 10.0.0.49 Up
```

After all BGP neighbors will come "Up," proceed to the next step.

Step 6

On the R1 router, verify IP routing table again.

```
R1# show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-
IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-
user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 18 subnets, 2 masks
C       10.0.0.0/28 is directly connected, Loopback1
L       10.0.0.1/32 is directly connected, Loopback1
B       10.0.0.16/28 [190/0] via 10.0.0.17, 00:01:28
O       10.0.0.17/32 [110/11] via 192.168.13.3, 01:29:04, Ethernet0/3
B       10.0.0.32/28 [190/0] via 10.0.0.33, 00:01:28
O       10.0.0.33/32 [110/11] via 192.168.12.2, 01:29:19, Ethernet0/2
B       10.0.0.48/28 [190/0] via 10.0.0.49, 00:01:28
O       10.0.0.49/32 [110/21] via 192.168.13.3, 01:28:40, Ethernet0/3
        [110/21] via 192.168.12.2, 01:28:40, Ethernet0/2
B       10.0.1.0/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.1.16/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.1.32/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.1.48/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.1.64/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.1.80/28 [100/0] via 172.16.11.11, 00:01:28
B       10.0.3.0/28 [190/0] via 10.0.0.49, 00:01:28
B       10.0.3.16/28 [190/0] via 10.0.0.49, 00:01:28
B       10.0.3.32/28 [190/0] via 10.0.0.49, 00:01:28
B       10.0.3.48/28 [190/0] via 10.0.0.49, 00:01:28
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       172.16.11.0/24 is directly connected, Ethernet0/0
L       172.16.11.1/32 is directly connected, Ethernet0/0
192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.12.0/24 is directly connected, Ethernet0/2
L       192.168.12.1/32 is directly connected, Ethernet0/2
192.168.13.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.13.0/24 is directly connected, Ethernet0/3
L       192.168.13.1/32 is directly connected, Ethernet0/3
O       192.168.24.0/24 [110/20] via 192.168.12.2, 01:28:50, Ethernet0/2
O       192.168.34.0/24 [110/20] via 192.168.13.3, 01:28:50, Ethernet0/3
```

Now, you can see that BGP routes AD values have changed per configuration.

Problems with BGP and IGP Interaction

There are several potential problems that might arise from [BGP](#) and [IGP](#) interaction.

If an IGP route is learned through EBGP, the EBGP route will take precedence.

- Potential causes include bad network design, routing problems, or denial-of-service attack.
- Protect IGP routes with inbound prefix-list filters at AS edges.
- Routers should never accept information about local subnets from an external source.

If routing information about the same [IP](#) prefix is learned via both [EBGP](#) and an IGP, the router will use the EBGP information. If an external [AS](#) is feeding the local AS with EBGP routes that actually should be local, routers within the AS will erroneously forward IP packets that are destined to those local networks out of the local AS.

There are several potential reasons for this behavior; the most common is that the remote AS is improperly configured or there is a [DoS](#) attack. To protect a local AS from this undesired behavior, network administrators should install inbound filters on all EBGP sessions to filter incoming routes and reject routing information about networks that are actually local to the AS.

Summary

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

- All core routers need external routers for proper packet forwarding.
- A recursive lookup is performed in BGP to resolve the forwarding path reference of the next-hop attribute.
- Packet forwarding to external destinations benefits from the high-speed convergence that IGP offers.
- The IGP should provide reachability toward BGP next-hop addresses only.
- IP packets could be erroneously forwarded out of the local AS if an external AS accidentally (or by intent: DoS) feeds the local AS with EBGP routes that should be local.

Monitoring and Troubleshooting IBGP in Transit AS

Overview

Introduction of a transit backbone into a [BGP](#) network can create unique troubleshooting challenges. This lesson introduces [IBGP](#) monitoring commands and troubleshooting techniques for solving the most common IBGP problems that you might encounter in a transit backbone. Common problems with IBGP, as discussed in this lesson, occur when IBGP sessions do not reach the Established state, when routing information that is received via IBGP is never selected, and when the best BGP route is never installed in the routing table.

Upon completing this lesson, you will be able to:

- Identify the Cisco IOS commands that are required to monitor IBGP operation
- Describe common IBGP configuration problems
- Explain how to troubleshoot IBGP session startup issues
- Explain how to troubleshoot IBGP route selection issues
- Explain how to troubleshoot IBGP synchronization issues

Monitoring IBGP

To display information about the [TCP](#) and [BGP](#) connections to neighbors, use the **show ip bgp neighbors** EXEC command. To display entries in the BGP routing table, use the **show ip bgp** EXEC command.

```
router> show ip bgp neighbors
```

- Displays whether a neighbor is an IBGP neighbor

```
router> show ip bgp
```

- Uses a special marker (i) for IBGP routes

```
router> show ip bgp prefix
```

- Displays whether the prefix is an IBGP route

Monitoring IBGP Example

This example shows the commands to monitor [IBGP](#).

```
R1# show ip bgp neighbors 10.0.0.49
BGP neighbor is 10.0.0.49, remote AS 1, internal link
  BGP version 4, remote router ID 10.0.0.49
  BGP state = Established, up for 00:47:57
  Last read 00:00:00, last write 00:00:37, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
<... output omitted ...>
```

- Internal BGP neighbor

```
R1# show ip bgp neighbors 172.16.11.11
BGP neighbor is 172.16.11.11, remote AS 100, external link
  BGP version 4, remote router ID 10.0.1.81
  BGP state = Established, up for 00:49:24
  Last read 00:00:37, last write 00:00:51, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
<... output omitted ...>
```

- External BGP neighbor

The **show ip bgp neighbors** command displays whether a router is running an IBGP (internal) or [EBGP](#) (external) session with a BGP neighbor. The "internal link" or "external link" phrases provide you with this information.

```
R1# show ip bgp 10.0.3.0
BGP routing table entry for 10.0.3.0/28, version 12
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  300
    10.0.0.49 (metric 21) from 10.0.0.49 (10.0.0.49)
      Origin IGP, metric 0, localpref 100, valid, internal, best
```

- Route that is received from internal BGP neighbor.

```
R1# show ip bgp 10.0.1.0
BGP routing table entry for 10.0.1.0/28, version 4
Paths: (1 available, best #1, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 1
  100
    172.16.11.11 from 172.16.11.11 (10.0.1.81)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

- Route that is received from external BGP neighbor.

The **show ip bgp** *prefix* command displays whether a BGP route was received from an IBGP (internal) or EBGP (external) neighbor. The word "internal" or "external" provides you with this information.

Common IBGP Problems

Troubleshooting the [BGP](#) configuration of a transit [AS](#) can be cumbersome, because there are a number of common pitfalls that you might encounter.

Three of the most common problems are the following:

- IBGP sessions do not start (do not reach the Established state).
- IBGP route is in the BGP table but is not selected.
- IBGP route is selected but is not entered in the routing table.

Troubleshooting IBGP Session Startup Issues

There are several approaches to [IBGP](#) session startup issues troubleshooting.

Symptom:

- IBGP session does not start.

Diagnosis:

- IBGP session is run between loopbacks, and **update-source** keyword is missing.

Verification:

- Use **debug ip tcp transactions**. You should see BGP sessions coming from unexpected IP addresses.

A common mistake when you are configuring IBGP sessions is to forget the **neighbor update-source** configuration command.

When you are configuring IBGP neighbors on the router, it is easy to remember to make a correct reference to the loopback interface of the remote router. But it is equally important to make sure that the correct source [IP address](#) of the outgoing [TCP](#) session is set. The peer router will not accept the session if the incoming source address does not match the peer router list of IBGP neighbors.

To verify that this issue is causing the problem, use the **debug ip tcp transactions** command. The output of the **debug ip tcp transactions** command should display TCP [SYN](#) packets coming from unexpected IP addresses on the receiving router and TCP sessions being reset with TCP [RST](#) packets on the sending (misconfigured) router.

Symptom:

- IBGP session does not start.

Diagnosis:

- Loopback interfaces are not reachable.

Verification:

- Do extended **ping** between loopback addresses to verify reachability.

An IBGP session between two routers can be established from the loopback interface of one router to the loopback interface of the other router only if the two routers can exchange [IP](#) packets using those addresses as source and destination. This exchange is possible only if the [IGP](#) carries the subnets that are assigned to each of the loopback interfaces.

When you are verifying reachability with the **ping** command, make sure that the ping packets are sourced from the loopback interface. Use an extended **ping** and explicitly refer to the IP address of the loopback interface to ensure that packets are sourced from the loopback interface.

Symptom:

- IBGP session does not start.

Diagnosis:

- Packet filters prevent establishment of BGP sessions.

Verification:

- Use **debug ip tcp transactions** and **debug ip icmp** to see whether the initial TCP SYN packets are rejected.

Packet filters can stop the [BGP](#) sessions. The path between the two BGP peer routers must be free from filters blocking the BGP traffic.

BGP runs on the well-known TCP port 179. Both routers will make connection attempts to that destination port. They will use a high-numbered TCP port as source. It is enough that one of the connection attempts succeeds. But for better performance during recovery from network failure, both attempts should have the possibility of succeeding. If both attempts do succeed, one of the connections will be brought down.

Troubleshooting IBGP Route Selection Issues

A router can use a [BGP](#) update to reach network destinations only if the next-hop address specified in the BGP update is reachable. A BGP update that refers to a next hop that is currently not reachable according to the routing table will be saved in the BGP table. However, the router cannot install this update in its routing table. If the next-hop address later becomes reachable, the BGP route will become a candidate route that the router could use for packet forwarding to that destination.

Symptom:

- An IBGP route is in the BGP table but is never selected as the best route.

Diagnosis:

- The BGP next hop is not reachable.

Verification:

- Use **show ip bgp prefix** to find the BGP next hop.
- Use **show ip route** to verify next-hop reachability.

To verify the next-hop reachability, check the BGP route in the BGP table by using the **show ip bgp prefix** command. The next hop is referred to as "inaccessible" if it is not currently reachable according to the routing table.

A common mistake is to forget to let the [IGP](#) announce the reachability of subnets that physically connect the local [AS](#) with a neighboring AS. The router uses these subnets to establish the [EBGP](#) session. The next hop that is received in an incoming BGP update will be the far end of the EBGP session. If all routers in the local AS do not have a path to that subnet, the next-hop address will be inaccessible.

You can prevent this problem by including the subnet that links the transit AS to neighboring autonomous systems in the IGP. To perform this action, use either the **redistribute connected** command or the **network** and **passive-interface** configuration commands.

Troubleshooting IBGP Synchronization Issues

In old [BGP](#) designs, redistribution between BGP and an [IGP](#) was a common practice, and these protocols had to be synchronized to ensure proper packet forwarding. In modern designs, redistribution is no longer used and synchronization has to be turned off. The default value is disabled synchronization.

Symptom:

- An IBGP route is selected as the best route but not entered into the IP routing table.

Diagnosis:

- BGP synchronization is not disabled.

Verification:

- Disable BGP synchronization, clear the BGP sessions, and re-examine the IP routing table after the BGP table becomes stable.

Routers with BGP synchronization enabled will not be able to install [IBGP](#) routes in the routing table or propagate them to other [EBGP](#) neighbors.

You can fix this problem by configuring **no synchronization** in the router BGP configuration.

Summary

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

- You can use the **show ip bgp neighbors** and **show ip bgp prefix** commands to monitor IBGP operation.
- There are a number of problems that can occur during IBGP session startup. You can use **debug ip tcp transactions** to see BGP sessions coming from unexpected IP addresses.
- It is important to include the subnet linking the transit AS to an external AS in the IGP to prevent the BGP next hop from being unreachable.
- Routers with BGP synchronization enabled will not be able to install IBGP routes in the routing table or propagate them to other EBGP neighbors. The BGP synchronization is disabled by default.

Module Summary

Overview

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

- Because all transit autonomous systems are required to carry traffic originating from or destined to locations outside of that AS, a degree of interaction and coordination between BGP and the IGP is necessary. You should take a special care to ensure consistency of routing information throughout the AS.
- Both EBGP and IBGP sessions forward BGP updates, but they do it in slightly different ways.
- An IGP is still needed inside a transit AS. The high-speed convergence offered by an IGP helps in the packet forwarding to external destinations.
- Configuring in a transit AS involves configuring IBGP neighbors, BGP synchronization, and IBGP sessions between loopback interfaces.
- The three common IBGP configuration problems concern session startup, route selection, and synchronization, and there are specific commands and procedures that you can use to solve those problems.

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *BGP Case Studies*
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html>
- Cisco Systems, Inc. *Using the Border Gateway Protocol for Interdomain Routing*
http://docwiki.cisco.com/wiki/Internetworking_Case_Studies
- Cisco Systems, Inc. *Troubleshooting TCP/IP*. "Troubleshooting IP Connectivity and Routing Problems."
<http://www.cisco.com/en/US/docs/internetworking/troubleshooting/guide/tr1907.html#wp1021148>
- Cisco Systems, Inc. *Configuring BGP*.
http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfbgp.html
- Cisco Systems, Inc. *Cisco IOS IP Routing Configuration Guide, Release 15M&T*
http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/15-mt/irg-15-mt-book.html

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. Why is IBGP a mandatory component of a transit AS? (Source: "Working with a Transit AS")

- It is the only feasible way to ensure that all routers in the AS have consistent external routing information.
- It eliminates the scalability issues of running an IGP within the transit AS.
- Running IBGP on all routers is the only way to satisfy the filtering requirements of the transit AS.
- An IGP is not capable of handling the potential routing loops in the transit AS.

1. IBGP neighbors must be directly connected. True or false? (Source: "Working with a Transit AS")

☐ true

☐ false

1. Why is redistributing BGP routes into an IGP for use in a transit backbone NOT recommended? (Source: "Working with a Transit AS")

- Redistribution removes all BGP attributes that are needed to ensure optimal routing within the transit AS.

- An IGP cannot enforce complex administrative policies and route selection rules.

- IGPs cannot scale to the demands that are presented by the number of routes on the Internet.

- IGPs are not stable when faced with a flapping network.

1. How is EBGp used in a transit AS? (Source: "Working with a Transit AS")

- as a means of transporting customer routes across the transit backbone
- to exchange routes between different autonomous systems and the transit AS
- to enhance scalability by transporting IGP routes for the transit AS
- as a means of injecting local routes into the transit backbone

1. The received EBGP information has to be translated and then forwarded to other BGP-speaking routers within the transit AS. True or false? (Source: "Working with a Transit AS")

☒ true

☐ false

1. All routers within a transit AS must have all external routes, received from any neighboring AS. True or false? (Source: "Working with a Transit AS")

☒ true

☐ false

1. What are the two key functions of a transit AS? (Choose two.)
(Source: "Working with a Transit AS")

- ☐ to filter out routes that do not belong to customers of the service provider
- ☐ to provide Internet connectivity to customers of the service provider
- ☐ to propagate routes between remote autonomous systems
- ☐ to route packets between remote networks

1. How are BGP routes sent across the transit backbone? (Source: "Working with a Transit AS")

- by redistributing BGP into an IGP and then back into BGP
- by using IBGP
- by establishing EBGP sessions between all routers in the transit backbone
- by redistributing connected routes at the edge of the transit backbone

1. Which two statements are true regarding the AS-path attribute as it relates to IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

☐ Each router in the AS appends its AS number to the AS path on outgoing BGP updates.

☐ The AS path inside an AS will be empty for routes originating inside a neighboring AS.

☐ The AS-path attribute is not used to detect routing loops inside an AS.

☐ The AS-path attribute is not modified within the AS.

1. What is the primary function of the IBGP multipath load-sharing feature? (Source: "Interacting with IBGP and EBGP in a Transit AS")

- to choose one IBGP path as the best path to a destination
- to choose multiple IBGP paths as the best paths to a destination
- to designate one path as the best path and advertise this best path to its neighbors
- to enable a router to handle all the traffic destined for a particular site

1. Which of the following statements about a BGP split horizon is accurate? (Source: "Interacting with IBGP and EBGP in a Transit AS")

Because of the BGP split horizon, any router can relay IBGP information within the AS.

BGP split horizon is useful only within autonomous systems.

Routing information loops within the AS are prevented by an IBGP split horizon.

With a BGP split horizon, routing information that is received through an IBGP session is forwarded to another IBGP neighbor.

1. Why is it recommended that loopback interfaces be used to form IBGP neighbor sessions? (Source: "Interacting with IBGP and EBGP in a Transit AS")

- reduces router memory resource requirements

- reduces router CPU resource requirements

- ensures IBGP session stability

- is more secure than using the physical interface

1. How is the BGP next-hop attribute processed over an IBGP connection? (Source: "Interacting with IBGP and EBGP in a Transit AS")

- The next-hop address is set to the address of the receiving router.

- The next-hop address is not modified over the IBGP session.

- The next-hop address is set to the IP address of the nearest EBGP peer.

- The next-hop attribute is set to the IP address of the nearest EBGP peer; if no external AS connection has been configured, the next hop is set to the default gateway that is configured on the router.

1. Which two statements are true of the full-mesh requirement in IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☐ The IBGP mesh must be a logical full mesh.
- ☐ A physical full mesh must be maintained within the IBGP AS.
- ☐ Because of BGP split horizon, no router can relay IBGP information within the AS.
- ☐ All routers within the AS must be directly connected to ensure correct delivery of BGP routing information.

1. Which three statements regarding the next-hop-self configuration in BGP are true? (Choose three.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☐ Changing the next-hop attribute might cause suboptimal routing.
- ☐ The configuration changes how the next-hop attribute is processed at edge routers.
- ☐ The configuration announces the local IP address as the BGP next hop in outgoing updates that are sent to the specified neighbor.
- ☐ The configuration removes the requirement for the IGP to carry reachability information for intra-AS destinations.

1. Why must all EBGP peers be reachable by all BGP-speaking routers within the transit AS? (Source: "Interacting with IBGP and EBGP in a Transit AS")

- BGP-speaking routers in a transit AS use the next-hop-self attribute to find their EBGP neighbors.

- EBGP peers in a transit AS use the length of the AS path to decide which BGP route to install in the routing table.

- When BGP routes are used in the routing table, the router checks how it would have reached the next-hop address, and it installs the BGP route with the same forwarding indication as for the route that is used to reach the next-hop IP address.

- All BGP peers do not need to speak to each other within a transit AS.

1. Which command is used to configure the router as the next hop for a BGP-speaking neighbor or peer group? (Source: "Interacting with IBGP and EBGP in a Transit AS")

☐ neighbor description

☐ neighbor remote as

☐ maximum-paths ibgp

☒ neighbor next-hop-sf

1. What are three differences between IBGP and EBGP sessions? (Choose three.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☐ Route selection rules slightly prefer IBGP routes.
- ☐ Routes that are learned from IBGP peers are not advertised to other IBGP peers.
- ☐ EBGP peers are directly connected, and IBGP peers are usually distant.
- ☐ By default, no BGP attributes are changed in IBGP updates.

1. When you are configuring the BGP neighbor session, what differentiates an EBGP neighbor from an IBGP neighbor? (Source: "Interacting with IBGP and EBGP in Transit AS")

- The keyword **internal** is at the end of the **neighbor** command.

- IBGP neighbors will have the same AS number specified.

- A description for the internal or external neighbor must be attached with the **neighbor description** command.

- Directly connected neighbors will automatically form an EBGP session.

1. What are two scalability tools that you can use to overcome the full-mesh requirement for IBGP sessions? (Choose two.) (Source: "Interacting with IBGP and EBGP in Transit AS")

- ☐ confederations
- ☐ floating static routes
- ☐ route reflectors
- ☐ disabling BGP synchronization

1. What are two negative ramifications of the full-mesh requirement that is imposed by IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in Transit AS")

- ☐ administrative difficulty of applying an AS-wide routing policy
- ☐ requirement to use next-hop-self for proper routing to external destinations
- ☐ large number of TCP sessions
- ☐ unnecessary duplication of routing traffic

1. It is not recommended to redistribute the information from external routers into the IGP which is used within the transit AS. True or false? (Source: "Forwarding Packets in a Transit AS")

☒ true

☐ false

1. What are two reasons why you must run IBGP on all routers within a transit backbone? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

- ☐ so routers can properly forward packets toward all external destinations
- ☐ to ensure that a full mesh exists among all routers in the AS
- ☐ to allow routers to properly process the BGP next-hop attribute
- ☐ because IGPs cannot scale large enough to handle redistribution of BGP routes

1. Routes learned via BGP do not have an outgoing interface associated with them in the routing table, so recursive lookup is performed to forward IP packets toward external destinations. True or false? (Source: "Forwarding Packets in a Transit AS")

☐ true

☐ false

1. If a transit backbone has IBGP running on all routers, what are two reasons why it is still necessary to use an IGP? (Choose two.)
(Source: "Forwarding Packets in a Transit AS")

☐ to provide routing information that is needed to establish the IBGP sessions

☐ to resolve next-hop references that are used in recursive routing

☐ so that BGP routes can be properly transported through the AS

☐ to provide user workstations with a network default gateway

1. Map routing protocol to its function. (Source: "Forwarding Packets in a Transit AS")

IBGP	exchanging information about how to reach internal destinations
IGP	exchanging information about how to reach external destinations

1. What is the AD of the following protocols? (Source: "Forwarding Packets in a Transit AS")

OSPF	120
RIP	115
IS-IS	110
IBGP	20
EBGP	200

1. What are two reasons why the AD is an important consideration for BGP network design? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

☐ The AD affects how routes are selected for use in the IP routing table.

☐ The AD controls how routing information is entered into the BGP table.

☐ If a route is advertised by both an IGP and through EBGp, the router will prefer the external route.

☐ The AD is not a large concern to BGP design, because the router will always choose the route that is advertised by the protocol that is best suited to reach the destination.

1. If a router learns the same information via EIGRP and IGP, which one will it use? (Source: "Forwarding Packets in a Transit AS")

☐ the one from IGP

☐ the one from EIGRP

1. With regard to recursive route lookups, what are two ways in which CEF is different from traditional Cisco IOS switching mechanisms such as route caching? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

- Traditional Cisco IOS switching mechanisms wait for the first packet to arrive before recursive lookup can take place.

- New entries in the IP routing table will trigger a recursive lookup in traditional Cisco IOS switching mechanisms.

- CEF prebuilds a complete IP forwarding table based on the IP routing table.

- CEF will build a FIB directly from the entries in the BGP table before any BGP packets arrive at the router.

1. Which administrative distance of BGP route cannot be changed?
(Source: "Forwarding Packets in a Transit AS")

☐ external distance

☐ internal distance

☐ local distance

☐ remote distance

1. Why is it important to disable BGP synchronization in a transit backbone? (Source: "Monitoring and Troubleshooting IBGP in Transit AS")

IGPs can support the routing requirements of full Internet routing, and hence synchronization is no longer necessary.

Because BGP redistribution into an IGP is no longer practical, enabling the synchronization feature is no longer applicable.

Synchronization requires all BGP transit routes to be explicitly mapped to an exit point, creating too much administrative overhead.

Synchronization requires BGP attributes to be properly mapped to IGP metrics for BGP routing across the transit backbone to function properly, creating too much overhead.

1. Which two steps are required when you use a loopback interface for IBGP peering sessions? (Choose two.) (Source: "Monitoring and Troubleshooting IBGP in Transit AS")

☐ Ensure that the loopback interfaces are reachable through an IGP.

☐ Ensure that the two neighbors are directly attached.

☐ Verify that each router has multiple physically redundant paths.

☐ Configure a neighbor statement with the **update-source** command.

1. Which Cisco IOS **show** command indicates that a BGP route is an IBGP route? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ **show ip route**

☐ **show ip route bgp**

☐ **show ip bgp**

☐ **show ip bgp internal**

1. Which Cisco IOS **show** command displays the information about the TCP and BGP connection to neighbors? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ **show ip bgp *prefix***

☒ **show ip bgp neighbors**

☐ **show ip bgp**

☐ **show ip bgp internal**

1. BGP runs on the well-known TCP port 169. True or False? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ true

☐ false

1. Which three of the following are the most common BGP implementation problems? (Choose three.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

- ☐ IBGP sessions do not reach the Established state.
- ☐ TCP window size is set incorrectly.
- ☐ Routing information that is received via IBGP is never selected.
- ☐ The best BGP route is never installed in the routing table.

1. What are three common situations that prevent IBGP sessions from starting? (Choose three.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ The IBGP session has been configured to peer to a loopback interface, but **update-source** has not been configured on the neighbor.

☐ An access-list filter is blocking access to TCP port 179.

☐ The IBGP session has been configured to peer to a loopback interface, but the loopback interface has not been administratively enabled with the **no shutdown** command.

☐ The IBGP session has been configured to peer to a loopback interface, but the interfaces are not reachable via the IGP.

1. Which common issue could prevent IBGP best routes from being inserted into the IP routing table? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

- failure to disable BGP synchronization

- failure to disable BGP split horizon

- lack of a route to the BGP next hop for the IGP

- failure to inject a default route into the IGP

1. Which two of the following statements about solving IBGP synchronization problems are accurate? (Choose two.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

- ☐ Routers with BGP synchronization enabled will be able to install IBGP routes in the routing table and propagate them to other EBGP neighbors.
- ☐ Routers with BGP synchronization enabled will not be able to install IBGP routes in the routing table or propagate them to other EBGP neighbors.
- ☐ The default value is to have synchronization enabled.
- ☐ The default value is to have synchronization disabled.

Answer Key

1. Why is IBGP a mandatory component of a transit AS? (Source: "Working with a Transit AS")

It is the only feasible way to ensure that all routers in the AS have consistent external routing information.

It eliminates the scalability issues of running an IGP within the transit AS.

Running IBGP on all routers is the only way to satisfy the filtering requirements of the transit AS.

An IGP is not capable of handling the potential routing loops in the transit AS.

1. IBGP neighbors must be directly connected. True or false? (Source: "Working with a Transit AS")

☐ true

☐ false

1. Why is redistributing BGP routes into an IGP for use in a transit backbone NOT recommended? (Source: "Working with a Transit AS")

- Redistribution removes all BGP attributes that are needed to ensure optimal routing within the transit AS.

- An IGP cannot enforce complex administrative policies and route selection rules.

- IGPs cannot scale to the demands that are presented by the number of routes on the Internet.

- IGPs are not stable when faced with a flapping network.

1. How is EBGp used in a transit AS? (Source: "Working with a Transit AS")

- as a means of transporting customer routes across the transit backbone
- to exchange routes between different autonomous systems and the transit AS
- to enhance scalability by transporting IGP routes for the transit AS
- as a means of injecting local routes into the transit backbone

1. The received EBGP information has to be translated and then forwarded to other BGP-speaking routers within the transit AS. True or false? (Source: "Working with a Transit AS")

☒ true

☐ false

1. All routers within a transit AS must have all external routes, received from any neighboring AS. True or false? (Source: "Working with a Transit AS")

☒ true

☐ false

1. What are the two key functions of a transit AS? (Choose two.)
(Source: "Working with a Transit AS")

- ☐ to filter out routes that do not belong to customers of the service provider
- ☐ to provide Internet connectivity to customers of the service provider
- ☒ to propagate routes between remote autonomous systems
- ☒ to route packets between remote networks

1. How are BGP routes sent across the transit backbone? (Source: "Working with a Transit AS")

- by redistributing BGP into an IGP and then back into BGP

- by using IBGP

- by establishing EBGP sessions between all routers in the transit backbone

- by redistributing connected routes at the edge of the transit backbone

1. Which two statements are true regarding the AS-path attribute as it relates to IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

☐ Each router in the AS appends its AS number to the AS path on outgoing BGP updates.

☐ The AS path inside an AS will be empty for routes originating inside a neighboring AS.

☐ The AS-path attribute is not used to detect routing loops inside an AS.

☐ The AS-path attribute is not modified within the AS.

1. What is the primary function of the IBGP multipath load-sharing feature? (Source: "Interacting with IBGP and EBGP in a Transit AS")

to choose one IBGP path as the best path to a destination

to choose multiple IBGP paths as the best paths to a destination

to designate one path as the best path and advertise this best path to its neighbors

to enable a router to handle all the traffic destined for a particular site

1. Which of the following statements about a BGP split horizon is accurate? (Source: "Interacting with IBGP and EBGP in a Transit AS")

Because of the BGP split horizon, any router can relay IBGP information within the AS.

BGP split horizon is useful only within autonomous systems.

Routing information loops within the AS are prevented by an IBGP split horizon.

With a BGP split horizon, routing information that is received through an IBGP session is forwarded to another IBGP neighbor.

1. Why is it recommended that loopback interfaces be used to form IBGP neighbor sessions? (Source: "Interacting with IBGP and EBGP in a Transit AS")

- reduces router memory resource requirements

- reduces router CPU resource requirements

- ensures IBGP session stability

- is more secure than using the physical interface

1. How is the BGP next-hop attribute processed over an IBGP connection? (Source: "Interacting with IBGP and EBGP in a Transit AS")

The next-hop address is set to the address of the receiving router.

The next-hop address is not modified over the IBGP session.

The next-hop address is set to the IP address of the nearest EBGP peer.

The next-hop attribute is set to the IP address of the nearest EBGP peer; if no external AS connection has been configured, the next hop is set to the default gateway that is configured on the router.

1. Which two statements are true of the full-mesh requirement in IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☒ The IBGP mesh must be a logical full mesh.
- ☐ A physical full mesh must be maintained within the IBGP AS.
- ☒ Because of BGP split horizon, no router can relay IBGP information within the AS.
- ☐ All routers within the AS must be directly connected to ensure correct delivery of BGP routing information.

1. Which three statements regarding the next-hop-self configuration in BGP are true? (Choose three.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☒ Changing the next-hop attribute might cause suboptimal routing.
- ☒ The configuration changes how the next-hop attribute is processed at edge routers.
- ☒ The configuration announces the local IP address as the BGP next hop in outgoing updates that are sent to the specified neighbor.
- ☐ The configuration removes the requirement for the IGP to carry reachability information for intra-AS destinations.

1. Why must all EBGP peers be reachable by all BGP-speaking routers within the transit AS? (Source: "Interacting with IBGP and EBGP in a Transit AS")

BGP-speaking routers in a transit AS use the next-hop-self attribute to find their EBGP neighbors.

EBGP peers in a transit AS use the length of the AS path to decide which BGP route to install in the routing table.

When BGP routes are used in the routing table, the router checks how it would have reached the next-hop address, and it installs the BGP route with the same forwarding indication as for the route that is used to reach the next-hop IP address.

All BGP peers do not need to speak to each other within a transit AS.

1. Which command is used to configure the router as the next hop for a BGP-speaking neighbor or peer group? (Source: "Interacting with IBGP and EBGP in a Transit AS")

☐ neighbor description

☐ neighbor remote as

☐ maximum-paths ibgp

☐ neighbor next-hop-sf

1. What are three differences between IBGP and EBGP sessions? (Choose three.) (Source: "Interacting with IBGP and EBGP in a Transit AS")

- ☒ Route selection rules slightly prefer IBGP routes.
- ☒ Routes that are learned from IBGP peers are not advertised to other IBGP peers.
- ☒ EBGP peers are directly connected, and IBGP peers are usually distant.
- ☒ By default, no BGP attributes are changed in IBGP updates.

1. When you are configuring the BGP neighbor session, what differentiates an EBGP neighbor from an IBGP neighbor? (Source: "Interacting with IBGP and EBGP in Transit AS")

The keyword **internal** is at the end of the **neighbor** command.

IBGP neighbors will have the same AS number specified.

A description for the internal or external neighbor must be attached with the **neighbor description** command.

Directly connected neighbors will automatically form an EBGP session.

1. What are two scalability tools that you can use to overcome the full-mesh requirement for IBGP sessions? (Choose two.) (Source: "Interacting with IBGP and EBGP in Transit AS")

- ☒ confederations
- ☐ floating static routes
- ☒ route reflectors
- ☐ disabling BGP synchronization

1. What are two negative ramifications of the full-mesh requirement that is imposed by IBGP? (Choose two.) (Source: "Interacting with IBGP and EBGP in Transit AS")

- ☐ administrative difficulty of applying an AS-wide routing policy
- ☐ requirement to use next-hop-self for proper routing to external destinations
- ☒ large number of TCP sessions
- ☒ unnecessary duplication of routing traffic

1. It is not recommended to redistribute the information from external routers into the IGP which is used within the transit AS. True or false? (Source: "Forwarding Packets in a Transit AS")

☒ true

☐ false

1. What are two reasons why you must run IBGP on all routers within a transit backbone? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

- ☒ so routers can properly forward packets toward all external destinations
- ☐ to ensure that a full mesh exists among all routers in the AS
- ☐ to allow routers to properly process the BGP next-hop attribute
- ☒ because IGPs cannot scale large enough to handle redistribution of BGP routes

1. Routes learned via BGP do not have an outgoing interface associated with them in the routing table, so recursive lookup is performed to forward IP packets toward external destinations. True or false? (Source: "Forwarding Packets in a Transit AS")

☐ true

☐ false

1. If a transit backbone has IBGP running on all routers, what are two reasons why it is still necessary to use an IGP? (Choose two.)
(Source: "Forwarding Packets in a Transit AS")

☒ to provide routing information that is needed to establish the IBGP sessions

☒ to resolve next-hop references that are used in recursive routing

☐ so that BGP routes can be properly transported through the AS

☐ to provide user workstations with a network default gateway

1. Map routing protocol to its function. (Source: "Forwarding Packets in a Transit AS")

IGP	exchanging information about how to reach internal destinations
IBGP	exchanging information about how to reach external destinations

1. What is the AD of the following protocols? (Source: "Forwarding Packets in a Transit AS")

IBGP	200
EBGP	20
OSPF	110
IS-IS	115
RIP	120

1. What are two reasons why the AD is an important consideration for BGP network design? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

☒ The AD affects how routes are selected for use in the IP routing table.

☐ The AD controls how routing information is entered into the BGP table.

☒ If a route is advertised by both an IGP and through EBGp, the router will prefer the external route.

☐ The AD is not a large concern to BGP design, because the router will always choose the route that is advertised by the protocol that is best suited to reach the destination.

1. If a router learns the same information via EIGRP and IGP, which one will it use? (Source: "Forwarding Packets in a Transit AS")

the one from IGP

the one from EIGRP

1. With regard to recursive route lookups, what are two ways in which CEF is different from traditional Cisco IOS switching mechanisms such as route caching? (Choose two.) (Source: "Forwarding Packets in a Transit AS")

☒ Traditional Cisco IOS switching mechanisms wait for the first packet to arrive before recursive lookup can take place.

☐ New entries in the IP routing table will trigger a recursive lookup in traditional Cisco IOS switching mechanisms.

☒ CEF prebuilds a complete IP forwarding table based on the IP routing table.

☐ CEF will build a FIB directly from the entries in the BGP table before any BGP packets arrive at the router.

1. Which administrative distance of BGP route cannot be changed?
(Source: "Forwarding Packets in a Transit AS")

- ☐ external distance
- ☐ internal distance
- ☐ local distance
- ☐ remote distance

1. Why is it important to disable BGP synchronization in a transit backbone? (Source: "Monitoring and Troubleshooting IBGP in Transit AS")

IGPs can support the routing requirements of full Internet routing, and hence synchronization is no longer necessary.

Because BGP redistribution into an IGP is no longer practical, enabling the synchronization feature is no longer applicable.

Synchronization requires all BGP transit routes to be explicitly mapped to an exit point, creating too much administrative overhead.

Synchronization requires BGP attributes to be properly mapped to IGP metrics for BGP routing across the transit backbone to function properly, creating too much overhead.

1. Which two steps are required when you use a loopback interface for IBGP peering sessions? (Choose two.) (Source: "Monitoring and Troubleshooting IBGP in Transit AS")

☒ Ensure that the loopback interfaces are reachable through an IGP.

☐ Ensure that the two neighbors are directly attached.

☐ Verify that each router has multiple physically redundant paths.

☒ Configure a neighbor statement with the **update-source** command.

1. Which Cisco IOS **show** command indicates that a BGP route is an IBGP route? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ **show ip route**

☐ **show ip route bgp**

☐ **show ip bgp**

☐ **show ip bgp internal**

1. Which Cisco IOS **show** command displays the information about the TCP and BGP connection to neighbors? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ **show ip bgp *prefix***

☒ **show ip bgp neighbors**

☐ **show ip bgp**

☐ **show ip bgp internal**

1. BGP runs on the well-known TCP port 169. True or False? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☒ true

☐ false

1. Which three of the following are the most common BGP implementation problems? (Choose three.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☒	IBGP sessions do not reach the Established state.
☐	TCP window size is set incorrectly.
☒	Routing information that is received via IBGP is never selected.
☒	The best BGP route is never installed in the routing table.

1. What are three common situations that prevent IBGP sessions from starting? (Choose three.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☒ The IBGP session has been configured to peer to a loopback interface, but **update-source** has not been configured on the neighbor.

☒ An access-list filter is blocking access to TCP port 179.

☐ The IBGP session has been configured to peer to a loopback interface, but the loopback interface has not been administratively enabled with the **no shutdown** command.

☒ The IBGP session has been configured to peer to a loopback interface, but the interfaces are not reachable via the IGP.

1. Which common issue could prevent IBGP best routes from being inserted into the IP routing table? (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ failure to disable BGP synchronization

☐ failure to disable BGP split horizon

☐ lack of a route to the BGP next hop for the IGP

☐ failure to inject a default route into the IGP

1. Which two of the following statements about solving IBGP synchronization problems are accurate? (Choose two.) (Source: "Monitoring and Troubleshooting IBGP in a Transit AS")

☐ Routers with BGP synchronization enabled will be able to install IBGP routes in the routing table and propagate them to other EBGp neighbors.

☒ Routers with BGP synchronization enabled will not be able to install IBGP routes in the routing table or propagate them to other EBGp neighbors.

☒ The default value is to have synchronization enabled.

☐ The default value is to have synchronization disabled.

Route Selection Using Policy Controls

Introduction

BGP enables traffic in Internet backbones to determine an optimal path to its destination across networks comprising more than one **AS**. Routes that are learned via BGP have properties that are associated with them that aid BGP in determining the best route to a particular destination. There are many instances in which the default BGP route selection does not match administrative or business policies. Likewise, redundant network designs often require enterprises to run BGP when they are connected to more than one ISP. In these situations, full BGP routing tables and default BGP route selection are not desirable.

This module provides information on how to connect Internet customers to multiple service providers. It introduces the need for filtering BGP updates and changing BGP route selection policies. In addition, this module describes different Cisco IOS mechanisms (AS path filters, prefix-lists, route-maps) available for BGP route filtering.

Upon completing this module, you will be able to:

- Influencing BGP route selection in a customer scenario where connections to multiple ISPs must be supported
- Configure BGP to influence route selection using AS path filters
- Define how to successfully configure BGP to influence route selection using prefix-list filters in a customer scenario in which connections to multiple ISPs must be supported
- Defines how to use outbound route filtering to minimize the impact of BGP routing updates on router resources in an operational BGP network
- Describe route-maps and how you can use them for BGP filtering.
- Define how to configure the soft reconfiguration feature to minimize the impact of expediting BGP policy updates in a typical BGP network

Using Multihomed BGP Networks

Overview

In some circumstances, it is important to have multiple paths to an [ISP](#). There are business and technical reasons to configure a [BGP](#) network in a multihomed configuration. Mission-critical applications often call for redundant network designs. When access to applications is provided over the Internet, enterprises typically use multihomed BGP networks to achieve their goals of high availability.

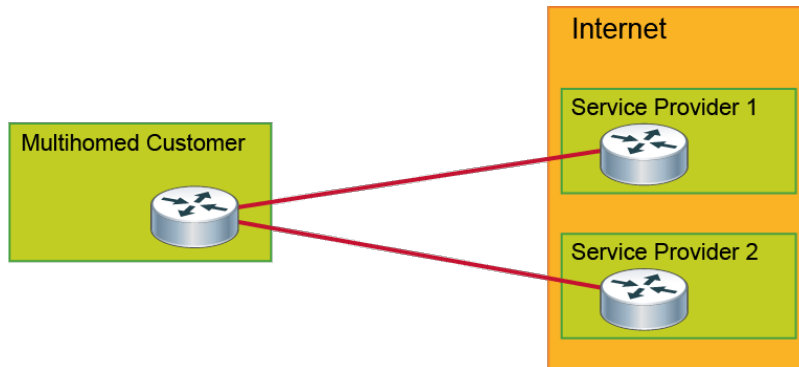
Full BGP routing tables and default BGP route selection might ordinarily be considered desirable characteristics of a network. However, the overhead of full BGP routing tables is not warranted in these situations. Furthermore, the default route selection in BGP often does not match the business and technical requirements for multihomed enterprise networks that use BGP.

Upon completing this lesson, you will be able to:

- List the business requirements for multihomed BGP networks in service provider environments
- Describe the technical requirements for multihomed BGP networks in service provider environments
- Explain the need for BGP policies that influence route selection in a multihomed BGP network
- Describe typical routing policies for multihomed BGP customers
- Explain the need to influence BGP route selection in a service provider environment
- Explain the need for BGP filters in a multihomed customer
- Explain the need for BGP filters in a service provider
- Explain the return traffic issue

Business Requirements for Multihomed BGP Networks

Companies who use web servers (or similar servers) and offer mission-critical business services over the Internet often like to have their networks redundantly connected to the Internet. When the companies calculate the expected loss of business because of an unexpected disconnection from the Internet, they may conclude that having two connections to the Internet is profitable. In such cases, the company may consider being a customer to two different [ISPs](#) or having two separate connections to one ISP.



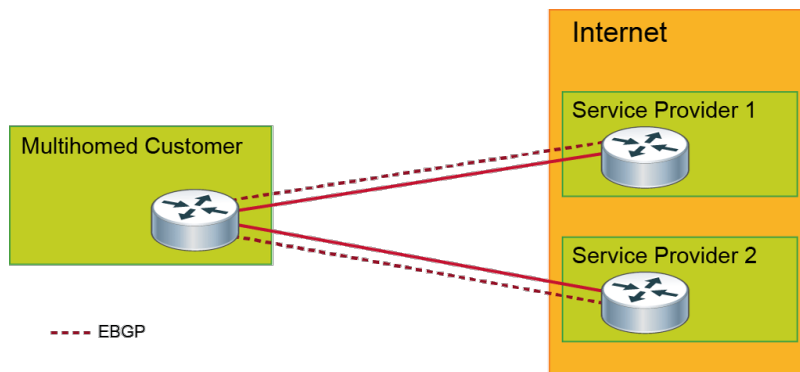
- Some customers need redundant Internet access for their mission-critical applications.
- Full redundancy is achieved only by connecting to two independent service providers.

With two connections to one single ISP, [BGP](#) is usually not required. This solution provides a backup for a link failure and router failure. However, this solution does not provide a backup for problems in the network of the ISP or the connection of the ISP to the rest of the Internet.

Full redundancy is achieved only by connecting to two independent ISPs. If one of the ISP networks loses its connection to the rest of the Internet, the customer will still reach the rest of the Internet via the other service provider. At the same time, the customer will still reach those users who are directly connected to the failing ISP via its direct connection.

Technical Requirements for Multihomed BGP Networks

The multihomed customer network must exchange [BGP](#) information with both [ISP](#) networks. Dynamic routing is required for full redundancy, and BGP is the only protocol available that can be used in this scenario.



- Multihomed customers have to run BGP with their ISPs.
- Multihomed customers usually need a public AS number and provider-independent address space.

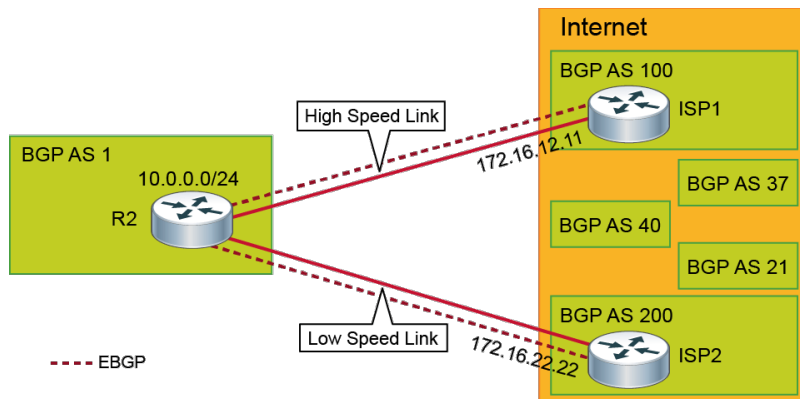
The customer must, in most cases, have its own public [AS](#) number and announce its own IP networks to both ISPs. The ISPs will propagate customer announcements to the rest of the Internet, and the customer will be seen as reachable via both ISP networks. The customer network also receives full Internet routing from both ISPs. This capability gives the customer network the opportunity to choose the best connection at that time to reach any destination on the Internet.

Most customers are not multihomed. They do not exchange BGP with their ISP. Instead, they use default routing to the ISP, and the ISP does static routing to the customer. ISPs use this fact to optimize the number of prefixes that they announce into the Internet. IP network numbers are usually assigned to customers from a range of IP networks that are delegated to the ISP. This situation means, in the ideal case, that all customers that are connected to one single ISP can have their IP networks summarized in a few BGP updates.

In the multihomed scenario, however, the ISP cannot benefit from IP network number assignment from the delegated range. The customer is connected to two different ISPs, and it is not obvious from which provider-assigned address space it should get the [IP addresses](#). The best solution is to do the assignment from a range completely independent of the providers, a provider-independent address space.

BGP Route Selection Without BGP Policies

The simple approach, which is illustrated in the figure, may be the source of many problems. By simply starting BGP sessions with both ISPs, and announcing the customer networks to both ISPs, the customer could experience difficulties as a result of the default behaviors of BGP. The following example illustrates problems that may occur in this environment.



```
router bgp 1
network 10.0.0.0 mask 255.255.255.0
neighbor 172.16.12.11 remote-as 100
neighbor 172.16.22.22 remote-as 200
```

- Customer configures two BGP sessions and announces its address space.

BGP Route Selection Without BGP Policies Example

- The BGP routes are selected based on AS-path length.
- The default BGP route selection does not always result in optimum routing.

```
R2# show ip bgp
BGP table version is 30, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  10.0.0.0/24      0.0.0.0              0         32768 i
*>  10.0.1.0/28      172.16.12.11         0         0 100 i
*>  10.0.2.0/28      172.16.22.22         0         0 200 i
*>  10.0.21.0/24     172.16.22.22         0         0 200 21 i
*                   172.16.12.11         0         0 100 37 40 2
1 i
*  10.0.37.0/24      172.16.22.22         0         0 200 21 40 3
7 i
*>                   172.16.12.11         0         0 100 37 i
*>  10.0.40.0/24     172.16.22.22         0         0 200 21 40 i
*                   172.16.12.11         0         0 100 37 40 i
```

The multihomed customer is connected to two ISPs: AS 100 and AS 200. The two ISPs are connected to the upstream ISPs: AS 21, AS 40, or AS 37.

The customer receives all routes from both service providers, giving redundancy. The default BGP route selection prefers the shortest AS path. If the AS path lengths are equal, BGP prefers the most stable route, or the route that is received from the peer with the lower router-ID.

In many cases, however, this route is not the most optimal way to reach all destinations. For example, the bandwidth that is available to reach the ISPs has not been taken into consideration. To change the route selection behavior, some BGP parameters must be configured to support more complex routing policies.

Multihomed Customer Routing Policies

Multihomed customers could require a number of routing policies, for example:

- One provider is primary; the other is backup.
- Traffic to direct customers of the ISPs goes direct; all other traffic goes through the primary provider.
- All traffic to a particular part of the world goes through one ISP.
- Traffic toward a specific destination goes through only one of the ISPs.

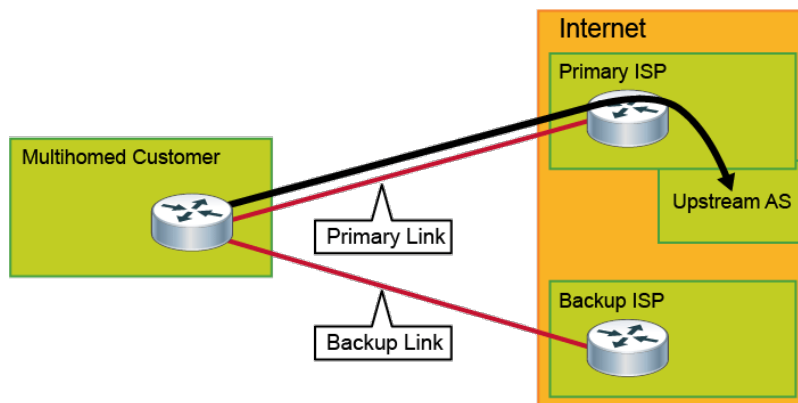
Depending on the circumstances, here are the different policies that a multihomed customer might require:

- One of the two [ISPs](#) can be considered the primary connection. This distinction can be the result of available bandwidth or commercial agreements. However, although one of the ISPs is considered the primary connection, some users may have direct connections to the secondary ISP. Therefore, going via the primary ISP to reach users that are connected to the secondary ISP may be suboptimal.
- Destinations in one part of the world may be reached more optimally via one of the ISPs, rather than via the other. This situation occurs because the two ISPs may have different infrastructures and peering agreements with other ISPs.

It is virtually impossible to establish a routing policy that gives optimal routing to every destination on the Internet. Optimization can be done only with the most common destinations in mind. This situation can result in specific rules on how to reach specific destination networks or the [AS](#).

Influencing BGP Route Selection

When one of the two [ISPs](#) is designated as a primary ISP and the other as a backup, [BGP](#) attributes must be configured to influence BGP route selection.

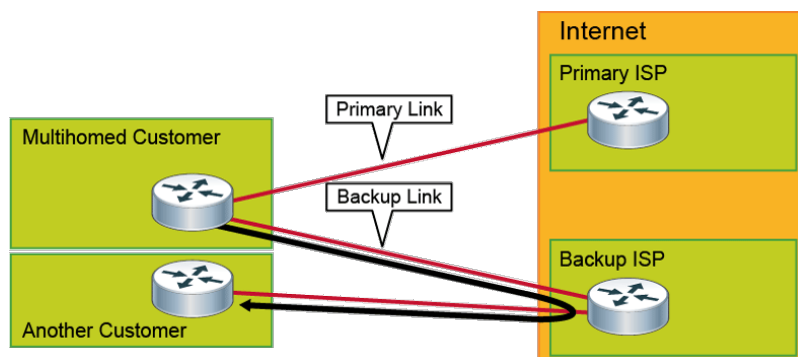


- Internet traffic always flows over primary ISP.
- Routes that are received from primary ISP should be preferred over routes that are received from backup ISP.
- A route selection tool is needed in BGP weights or local preference.

If both ISP connections terminate in one single customer router, all routes that are received from the primary ISP can be assigned a BGP weight. A higher weight indicates a more preferred path.

However, the weight value is local to one router. The weight value is not shared between routers. If one ISP connection terminates in one of the customer routers and the other ISP connection terminates in another, the two customer routers must agree on which link to use. Using local preference instead of weight can cause this situation. All routes that are received from the primary ISP over the primary link are assigned a local preference value, which is higher than the default value of 100. The customer router that receives the routes from the primary ISP completes the assignment and communicates the information to the other routers within the [AS](#) of the customer.

When using either weight or local preference, the customer AS reaches all destinations on the Internet via the primary link as long as it is available. It reaches those destinations within the AS of the secondary ISP. If there is a link failure or failures within the network of the primary ISP, some of the routes, or all of the routes, will no longer be received over the primary link. In that case, the AS of the customer no longer sees those destinations as reachable over the primary link. The only remaining choice is the backup link. Therefore, the customer network uses the backup link only to reach destinations that are not reachable over the primary link.



- Internet traffic flows over primary ISP; traffic to customers of backup ISP goes direct.
- Route selection has to be performed based on AS numbers in the AS path.

In most cases, it is optimal to reach other customers who are connected to the backup ISP via the backup link, compared with reaching them via the primary link.

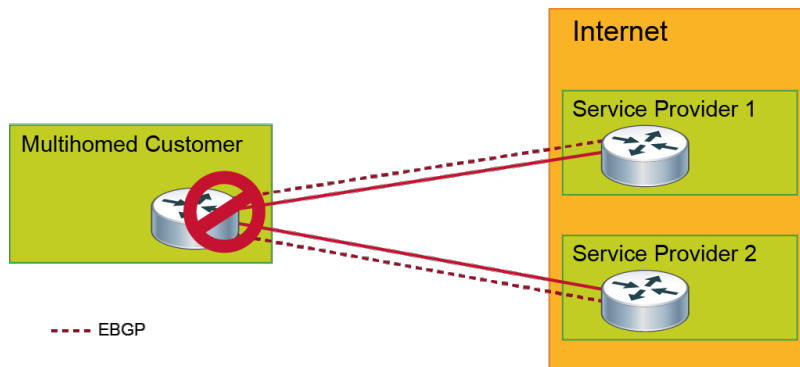
The routing policy, where routes are blindly preferred if they are received on the primary link, can easily be modified to use the backup link for destinations in the AS

of the backup ISP. On the router, filtering tools can be configured to select routing information that is based on the content in the AS path attribute. Those routes, with an AS path attribute matching specific selection criteria, can be assigned an even higher weight or local preference.

This approach results in a routing policy that gives precedence to reaching destinations within the AS of the primary ISP and within all ASs upstream of the primary ISP over the primary link. Destinations within the AS of the backup ISP receive precedence over the backup link.

Transit Traffic Issue

When BGP has selected the best path, the router advertises the information to all neighboring autonomous systems, except for the session that the path was received on. This functionality is called "BGP split-horizon," which prevents near-range routing loops. This situation causes the customer AS to become a transit AS between the two ISPs, and should be avoided.



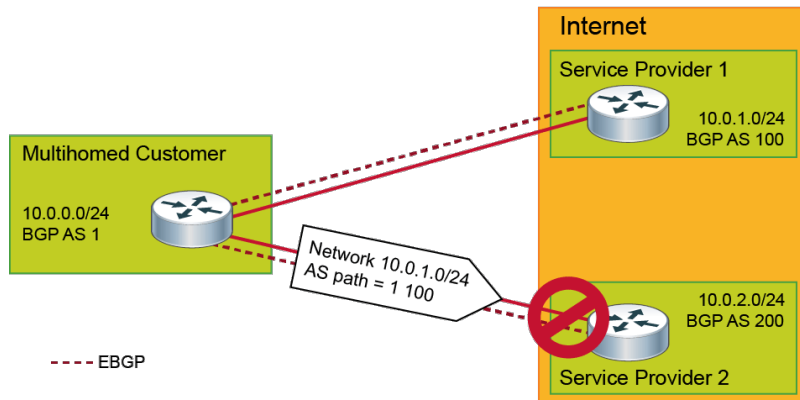
- Customers could become a transit AS for the service providers.
- **Requirement:** Do not propagate provider routes to other providers.

Most customers do not intend to create transit traffic between ISP networks. The access lines to the ISPs are not suited to carry this volume of traffic, and the customer certainly does not want the transit traffic to consume the bandwidth.

The solution to this problem is to filter the outgoing information to both ISPs. Filtering of routing information is performed based on the content of the AS path attribute that is assigned to every BGP route. Only routes having an AS path attribute that indicates that they are sourced by the AS of the customer are allowed to be sent to either of the two ISPs.

Routing Update Reliability Issue

Without some sort of filtering, BGP routing information that the AS of the customer creates can potentially be propagated all over the Internet. In this way, the customer can inject erroneous information into the Internet routing tables.



- Customers running BGP could announce any route to the SPs.
- **Requirement:** Service providers have to filter IP prefixes in incoming updates.

Customers are much less experienced in avoiding these kinds of problems than are service providers. There is much more risk of errors being introduced when a customer is assigned its own AS and uses BGP with the ISP, as compared with the single-homed scenario in which the ISP has sole responsibility to announce BGP routes to the rest of the Internet.

The ISP can stop almost all of the Internet problems that a customer can cause by improperly configuring its BGP. The ISP should also filter all incoming information from the customer and accept only what is supposed to arrive. The ISP should discard anything outside strict limits. In this way, the ISP prevents the propagation of erroneous information to the rest of the Internet.

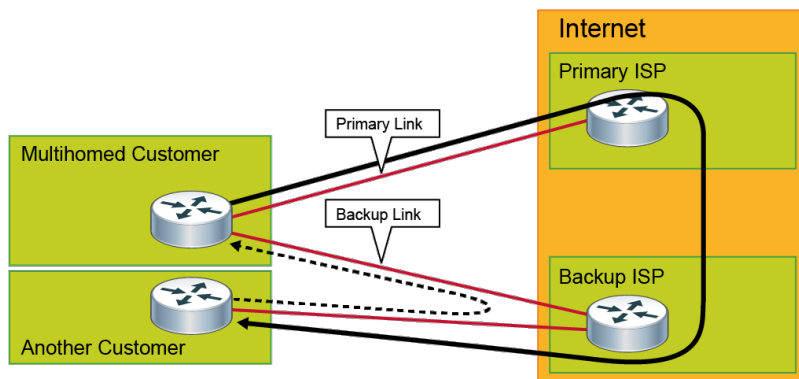
The ISP can maintain a list of the IP network numbers that the customer is announcing and filter out any other route. If this approach is not possible because of the volume of those lists, the ISP should at least be able to filter out the most obvious erroneous announcements.



Private addresses, according to [RFC 1918](#), should never be announced to the Internet.

Return Traffic Issue

The customer can easily define a policy about how to send outgoing IP packets on the correct link. It is much harder to influence the neighboring [AS](#) about how to direct the IP packets coming into the customer network.



- Customers can influence only their outgoing traffic, not the return traffic.
- Return traffic can take any path—backup ISP must also perform proper route selection.

A customer that creates a routing policy in which one of the two [ISPs](#) is always preferred may see the return traffic arriving on the backup link. This situation means that the customer has configured the weight or local preference to make sure that all outgoing traffic is leaving the customer AS over the primary link. However the backup ISP does not have any such configuration. Therefore, return traffic enters the customer AS by using the shortest AS path as its selection criterion.

The best way to solve this problem is for the customer to ask the backup ISP to change its routing policy. The change should cause the backup ISP to prefer reaching the customer AS via the AS of the primary ISP. The backup ISP must implement this change in its own AS.



Sometimes the backup ISP administrator might be reluctant to change the configuration for a single customer. In this case, the customer should use another [BGP](#) feature, the AS path prepending tool, to influence the selection of the primary or backup link by lengthening the AS path of routes that are sent to the backup provider.

Summary

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

- Some customers need redundant Internet access for their mission-critical applications and address this need by having two separate connections to one ISP or implementing a multihomed configuration (connecting to two different Internet service providers).
- The multihomed customer network must exchange BGP information with both ISP networks. Dynamic routing is required for full redundancy, and BGP is the only protocol available that can be used in this scenario.
- An approach to multihoming that is too simple can be a source of problems. Starting BGP sessions and announcing customer networks to multiple ISPs by using the default behavior of BGP may not result in optimal routing.
- Depending on the circumstances, a multihomed customer may require different policies. For example, one of the two ISPs being considered the primary connection or reaching destinations in one part of the world more optimally via one of the ISPs. Optimization should be done with the most common destinations in mind, resulting in specific rules on how to reach specific destination networks or the AS.
- In BGP route selection, a routing policy may be created that gives precedence to reaching destinations within the AS of the primary ISP and all upstream ASs over the primary link. All destinations within the AS of the backup ISP can be reached over the backup link.
- When BGP has selected the best path and the information has been propagated to all neighboring autonomous systems, the customer AS may become a transit AS between the two ISPs. The customer must avoid this situation by using BGP filters.

Employing AS Path Filters

Overview

In network implementations that require connections to multiple [ISPs](#), you will typically use [AS](#) path filters to influence [BGP](#) route selection. It is important for you as a network administrator to understand the syntax of an AS path regular expression. It is also important to understand how string-matching operators function when they are using AS path regular expressions to match BGP routes.

BGP allows connectivity between multiple ISPs for redundancy and scalability. You can employ AS path filters to remedy the problems that are associated with the various connectivity methods that are used within BGP. This lesson explains the methods that are used to implement BGP AS path filters.

Upon completing this lesson, you will be able to:

- Identify network scenarios in which you must support connections to multiple ISPs and in which AS path filters can be used to influence route selection
- Describe the function of an AS path regular expression
- Explain how string-matching operators function when you are using AS path regular expressions to match BGP routes
- Identify where you can apply an AS path filter when configuring a router to influence route selection
- Identify the Cisco IOS commands that are required to configure AS path filters to influence route selection
- Identify the Cisco IOS commands that are required to monitor the operation of configured AS path filters

AS Path Filtering Scenarios

Several scenarios require filtering and selection of routing information, which is based on the content of the [AS](#) path attribute. Each [BGP](#) route must have an AS path attribute. It is a well-known mandatory attribute and must therefore be present in each BGP update.

Several scenarios require BGP route-filtering which is based on the AS path:

- Announce only local routes to the ISP—AS path needs to be empty
- Select routes that are based on a specific AS number in the AS path
- Accept routes for specific AS only from some BGP neighbors

AS path filters use regular expressions.

Using selection criteria that are based on the AS path attribute, a router can identify a set of specific routes from the total set of routes that it receives. Those routes where the AS path contents match the criteria are selected. Routes that do not match the criteria are not selected.

The AS path is a sequence of numbers. Each number indicates an AS. When a route is sourced by a **network** command in a BGP process or redistribution into a BGP process, the AS path attribute is created and left empty. Each time the egress router advertises the route to another AS, it modifies the AS path attribute. Egress router prepends its AS number to the AS path attribute.

While a newly sourced route is still within the AS in which it was created, the AS path is empty. Assume that the AS has a requirement to filter out all but the routes that are local to itself before sending them to a neighboring AS. The AS will permit sending of the routes with the empty AS path and will deny all others.

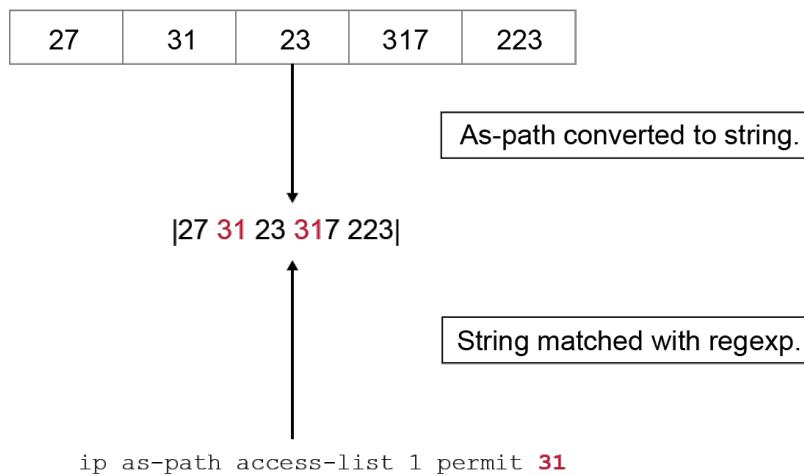
Routers can also filter incoming routes that are based on their AS path attributes. Some destination autonomous systems should not be received from a certain neighbor. Therefore, the routes matching that AS in the AS path can be filtered on the receiving router in case they are accidentally sent.

Selection that is based on the AS path is also a tool that you can use when changing the weight or local preference attributes only for some destination ASs.

When routers filter BGP updates based on the content of the AS path attribute, they use regular expressions. Regular expressions are commonly found in the [UNIX](#) environment and also in some Microsoft Windows-based applications. Regular expressions are a string-matching tool. A regular expression consists of a string of characters. Some of these characters have special meanings, such as functioning as wildcards and operators. Some of these characters simply mean themselves, for example, A to Z, a to z, or 0 to 9. A regular expression is said to match a string if the ordinary characters and the applied meaning of the special operator characters can be translated into the matched string. When a regular expression matches, the selection test is said to be true. If it does not match, the test is false.

AS Path Regular Expressions

The [AS](#) path attribute that is carried with all [BGP](#) routes in a BGP update is a very compact binary encoding of a sequence of integer numbers. It is not a sequence that can be tested by using a regular expression.



Cisco IOS software internally translates the binary encoding into a character string. Each AS number in the sequence is converted into a string using decimal representation. The space character separates each AS number in the AS path attribute. The router applies the regular expression test to this internally created character string.

Characters in a regular expression that are not assigned a specific operation match themselves. The regular expression "31" matches all occurrences of the character "3" followed by the character "1" in the AS path. In this example, "31" matches at two occurrences. One occurrence is sufficient to make the test true. No occurrence means that the test failed.

String Matching

How does string matching functions when you are using AS path regular expressions to match BGP routes?

Regular Expressions

A string of characters in a regular expression matches any equivalent substring in the AS path.

How many times does "31" match?

```
| 213 317 2316 31 |
```

Answer: Three times

```
| 213 317 2316 31 |
```

The regular expression "31" will match any occurrence of "3" followed by "1" regardless of the characters immediately preceding the "3" and immediately following the "1." Therefore, "31" will match an occurrence of "3" and "1" in the middle of an [AS](#) number.

The regular expression "31" matches the AS path string "213 317 2316 31" three times, because "31" matches a part of "317," "2316," and "31."

Alternatives

Expression *expr1**expr2* matches the string if either subexpression matches the string.

How many times does "21|31" match?

```
| 213 317 2316 31 |
```

Answer: Four times

```
| 213 317 2316 31 |
```

The character "|" (vertical bar) has a special meaning. It is an operator that means "or." The regular expression "21|31" matches the sequence of "2" followed by "1" or the sequence of "3" followed by "1." Therefore, this sample regular expression will match a two-character sequence: the "21" or the "31."

The regular expression "21|31" matches the AS path string "213 317 2316 31" four times, because "21" matches a part of "213" and "31" matches a part of both "317" and "2316" and also "31."

Ranges and Wildcard Characters

A range of characters matches any single character in the range.

Example: "[1234]" or "[1-4]"

Dot "." matches any single character

How many times does "[1-3].[34]" match?

```
| 213 317 2316 31 |
```

Answer: Twice

```
| 213 317 2316 31 |
```

```
| 213 317 2316 31 |
```

The pair of brackets "[" and "]" has a special meaning. Brackets surround a set of characters of which any one matches. The set of characters is either expressed as the list of the matches (for example, "[1234]") or the sequence with the starting

character, a hyphen, and the ending character (for example, "[1-4]"). Both examples match one single character, which must be any one in the set of the four characters "1," "2," "3," and "4."

The character "." (a dot) matches any single character. Small regular expressions can be combined into a larger expression. Such a combination is matching if all of the parts match one after the other. The sample regular expression "[1-3].[34]" matches a sequence of three characters, of which the first must be either "1," "2," or "3." The second character can be any character, and the third must be either "3" or "4."



The space character delimiting two AS numbers is just a character. The dot (".") for example, matches this character.

The regular expression "[1-3].[34]" matches the AS path string "213 317 2316 31" twice. Initially, it matches "213." The leading "[1-3]" matches the leading "2." The dot, which matches any character, matches the "1," and "[34]" matches the trailing "3." The regular expression also matches in "213 317 2316 31." The match is a little harder to see, because the dot (".") matches the space character between "213" and "317."

Matching Delimiters

^	Matches beginning of string
\$	Matches end of string
_	Matches any delimiter (beginning, end, white space, tab, comma)

| 213 317 218 31 731 |

Does "^21" match?

Answer: Yes

| 213 317 218 31 731 |

| 213 317 218 31 731 |

Does "31\$" match?

Answer: Yes

| 213 317 218 31 731 |

Does "_31_" match?

Answer: Yes

| 213 317 218 31 731 |

A character string must have a start and an end. The character with the special meaning "^" matches the beginning of a string. Because all strings have a beginning, the "^" character matches all strings. However, the "^" character is used to position the part of the regular expression that follows. The character following the "^" character must be the first character of the string; otherwise, that character would not match the beginning of the string.

The special character "\$" is used analogously, but it means the end of the string. The character preceding the "\$" must be the last character in the string; otherwise, the "\$" does not match the end of the string.

The underscore ("_") matches any delimiter. The space character between two AS numbers is an example of a delimiter. The beginning of the string and the end of the string are also considered delimiters. Other delimiters are the tab and the comma (","). The underscore ("_") is used to ensure that the desired AS number is found in an AS path string but not as part of some other AS number. For example, the regular

expression "31" will match the AS number string "317," but the regular expression "_31_" will not. Both "31" and "_31_" will match the AS number string "31."

The regular expression "^21" can match the AS path string "213 317 218 31 731" only one time because there is only one beginning of the string. The regular expression "^21" matches only if the string starts with the sequence "21," which it does.

The regular expression "31\$" can match the AS path string "213 317 218 31 731" only one time because there is only one end of the string. The regular expression "31\$" matches only if the string ends with the sequence "31," which it does.

The regular expression "_31_" can, in theory, match an AS path string several times. However, in this case, when matched against the string "213 317 218 31 731," the regular expression "_31_" matches only the AS number "31" in the AS path.

Grouping

Parentheses can be used to group smaller regular expressions into larger expressions.

How many times does "(213|218)_31" match?

```
| 213 317 1218 316 31 |
```

Answer: Twice

```
| 213 317 1218 316 31 |
```

Complicated expressions must sometimes be grouped with parentheses, "(" and ")." This feature can be useful in the following example. You are searching for a sequence of two or more AS numbers of which the first can match any of the specific AS (in this example, "213" or "218"). However the last must be a specific AS ("31" in this example). If the parentheses were not used here, the expression would match either the single AS "213" or the sequence of the two (218 31).

The regular expression "(213|218)_31" matches the AS path string "213 317 1218 316 31" twice. The first match is "**213 317** 1218 316 31." The second match is "213 317 **1218 316** 31."

Special Characters

To use the special character as single-character patterns, remove the special meaning by preceding each character with backslash "\"

How do you match "AS 213" in the beginning of the string?

```
| (213 317) 1218 316 31 |
```

Answer:

```
^\(213_
```

Sometimes the target string that you are trying to match with a regular expression contains some of the characters that also have special meanings in the regular expression. To match these characters in the target string, use the backslash "\" together with the character in the regular expression.



This type of regular expression syntax is used for matching AS path strings inside a [BGP](#) confederation. A confederation is used to eliminate the scaling problem of full-mesh [IBGP](#) by splitting the AS into smaller regional autonomous systems. The example shows that 213 and 317 were part of a confederation by its use of "(" and ")."

Repeating Operators

*	Matches zero or more atoms
?	Matches zero or one atom

+	Matches one or more atoms

An atom is a single character or a grouping.

How do you match AS sequences "23 45" and "23 78 45" in a single regular expression?

Answer:

<code>_23(_78)?_45_</code>

The special characters, star (asterisk), "*", question mark, "?", and plus, "+", all apply repetition of the expression that immediately precedes them.

The star (asterisk), "*", means that the expression that immediately precedes it is repeated zero or more times. The expression may not be there, but it may also be there any number of times. The expression "1*" will match a sequence of no characters or a sequence of any number of the character "1."

A question mark means, that the expression that immediately precedes it is repeated zero or one time. The expression may not be there, but it may also be there once. The expression "1?" will match a sequence of no characters or the single character "1."

The plus sign ("+") means that the expression that immediately precedes it is repeated one or more times. The expression must be there at least once. The expression "1+" will match a sequence of one or more of the characters "1."

String Matching Example

<code>_100_</code>	Going through AS 100
<code>^100\$</code>	Directly connected to AS 100
<code>_100\$</code>	Originated in AS 100
<code>^100_.</code>	Networks behind AS 100
<code>^[0-9]+\$</code>	AS paths one AS long
<code>^([0-9]+)_1*\$</code>	Prepending performed in neighboring originating AS
<code>^\$</code>	Networks that originate in local AS
<code>.*</code>	Matches everything

Regular expressions can be arbitrarily complex. However, most searching is accomplished using regular expressions similar to the regular expressions found in the following examples:

- If you are searching for all routes that have AS 100 in their AS paths, the regular expression to use is `"_100_"`.
- If you are searching for all routes that are sourced in your directly connected neighboring AS 100, the regular expression to use is `"^100$"`. Use an expression with both a carat ("^") and dollar sign (\$) present when you are searching for an exact match.
- If you are searching for all routes that are sourced in AS 100, which is not necessarily a directly connected neighboring AS, the regular expression to use is `"_100$"`. The dollar sign (\$) indicates that the AS path must end with AS 100. Meaning that the route was sourced in AS 100. The underscore ("_") is used to make sure that it is AS 100 at the end of the string and not, for example, AS 2100.
- If you are searching for all the routes that are reachable behind AS 100, the regular expression to use is `"^100_."` The carat (^) indicates that the AS path must start with AS 100. The underscore ("_") is used to make sure that this number is not matching with, for example, AS 1001. The dot (".") is used to indicate that the AS path does not end with AS 100, and that there must be something following AS 100.
- If you are searching for all routes that are sourced in any AS directly neighboring your AS, the regular expression to use is `"^[0-9]+$"`. The "[0-9]" part means any

digit. The plus sign ("+") repetition character means one or more times. Therefore, the combination "[0-9]+" means a sequence of one or more of digits. The carat ("^") and dollar sign ("\$") mean the beginning and the end of the string. Therefore, the string may consist only of a sequence of one or more digits.

- If you are searching for all routes that are sourced in any AS directly neighboring your AS, and possibly performing AS path prepending (multiplication of a directly connected AS number), the regular expression to use is "^([0-9]+)(_\\1)*\$." The expression in the first set of parentheses matches any AS number. The parentheses store the value of the matched AS, and the second part of the regular expression, including a variable, then recalls it. The variable "\\1" is put into parentheses for the purpose of the multiplier operator "*." I meaning that this part can match any number of successive occurrences of the same AS number that the "[0-9]+" expression matched. For example, this regular expression matches AS paths "99 99 99," "200," "101 101," or "5 5 5 5," but it does not match the AS path "101 99."
- The combination "^\$" means an empty string and is used when you are searching for all routes that are sourced in the local AS.
- Sometimes a search is made to select a few specific routes and do something special with them, while the rest of the routes will be handled in a different way. To search for all routes, regardless of the content of their AS path attribute, use the regular expression ".*". The dot (".") matches any single character. The repetition character, star (asterisk), "*", means that the match should be repeated zero or more times. Thus, the combination (".*") matches any string.

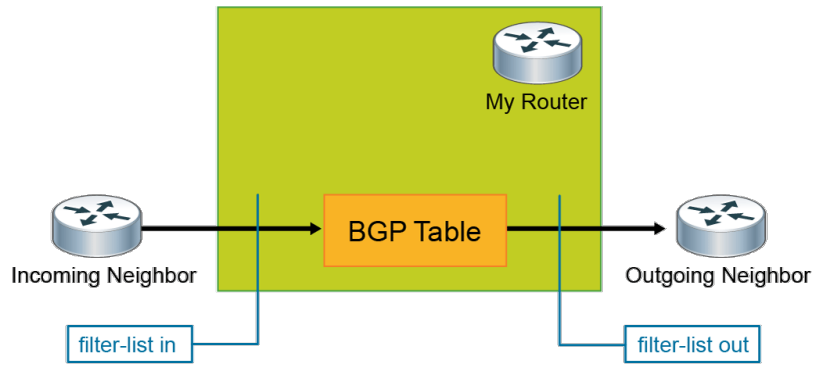
Commonly Used Characters in Expressions Example

.	Any single character, including a space
*	Zero or more sequences of pattern
+	One or more sequences of pattern
?	Zero or one occurrence of pattern
^	Beginning of string
\$	End of string
_	Match any delimiter (including beginning, end, space, tab, comma)
\	Remove special meaning of character that follows
[]	Match one character in a range
	Logical OR

The figure lists the most commonly used characters in expressions.

Applying AS Path Filters

AS path filters that are configured on a router select those routes that are allowed.



Routes that are selected behave as described here:

- The selected routes enter the local [BGP](#) table when the selection is applied on the incoming routes from a neighbor; routes that are not selected are silently dropped.
- The selected routes are transmitted to the neighbor when the selection is applied on the outgoing routes to the neighbor. Routes that are not selected are used locally but are never sent to the neighbor.

Configuring BGP AS Path Filters

AS path access list creates an AS path filter. This access list is applied to a set of routes from which to select a subset. Routes that the access list permits are included in the subset, and those routes that are denied are not included. As in all access lists, the candidate to be permitted or denied membership in the subset is tested against all the lines in the access list. The order of testing is the order in which the list is configured. The first match indicates "permit" or "deny," as specified. If the end of the access list is reached without any explicit match, the candidate is implicitly denied.

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

- Configures AS path access-list

```
router(config-router)# neighbor ip-address filter-list as-path-filter {in|out}
```

- Configures inbound or outbound AS path filter for specified BGP neighbor

The test by the AS path access list is performed by using regular expressions that are applied on the AS path attribute of the route.

The access list can, for example, be applied on the routes received from, or sent to, a specific [BGP](#) neighbor.

ip as-path access-list

To define a BGP AS path access-list, use the **ip as-path access-list** global configuration command:

```
ip as-path access-list access-list-number {permit | deny} as-regular-expression
```

To disable use of the access-list, use the **no** form of this command.

```
no ip as-path access-list access-list-number {permit | deny} as-regular-expression
```

Syntax Description

Parameter	Description
<i>access-list-number</i>	Integer from 1 to 199 that indicates the regular expression access-list number.
permit	Permits access for matching conditions.
deny	Denies access to matching conditions.
<i>as-regular-expression</i>	AS in the access-list using a regular expression (See the "Regular Expressions" appendix in the <i>Cisco IOS Dial Services Command Reference</i> for information about forming regular expressions.)

neighbor filter-list

To set up a BGP filter, use the **neighbor filter-list** router configuration command:

```
neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}
```

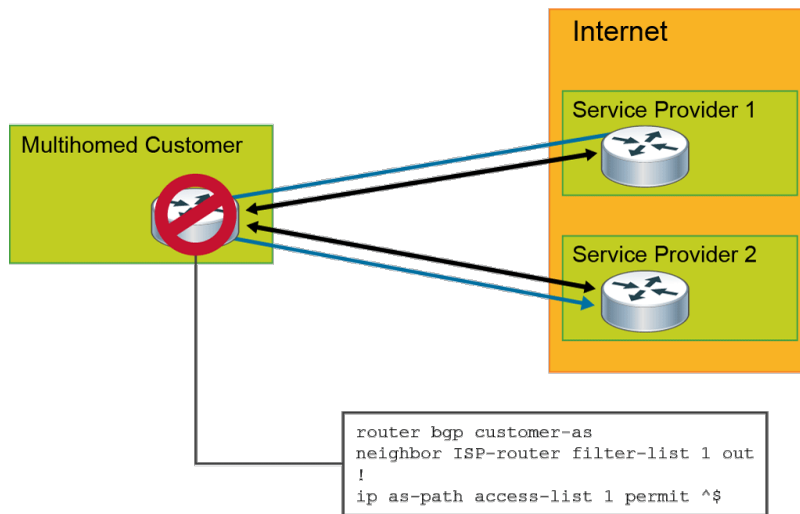
To disable this function, use the **no** form of this command:

```
no neighbor {ip-address | peer-group-name} filter-list access-list-number {in | out}
```

Syntax Description

Parameter	Description
<i>ip-address</i>	IP address of the neighbor.

Parameter	Description
<i>peer-group-name</i>	Name of a BGP peer group.
<i>access-list-number</i>	Number of an AS path access-list. Define this access-list with the ip as-path access-list command.
in	Access-list to incoming routes.
out	Access-list to outgoing routes.



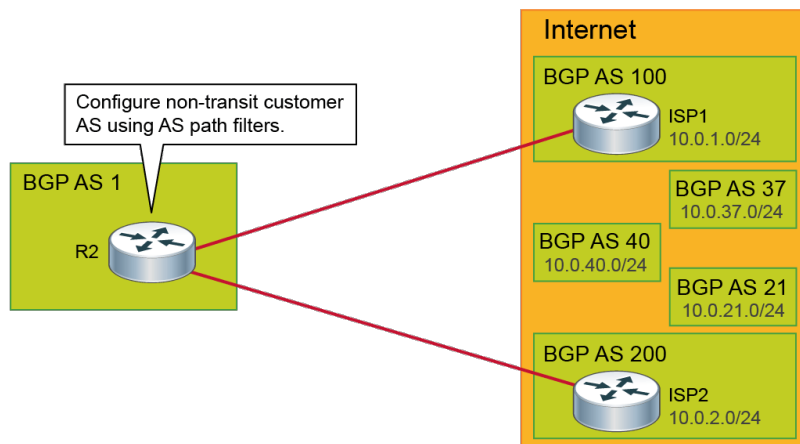
Multihomed customers do not want to act as a transit AS between their service providers. The customer avoids this situation by not transmitting all its routes to its service providers. The service provider sends IP packets to the customer only if the destination addresses match one of the routes that the customer has sent by BGP to the service provider. By making sure that only locally sourced routes are sent, the customer avoids receiving IP packets for destinations outside its own AS.

Within the customer AS, the locally sourced routes have empty AS paths. The regular expression "^\$" matches the empty strin. The command **ip as-path access-list 1** permits only the routes that are locally sourced and implicitly denies the rest. By applying this filter-list on outgoing information to all neighbors, the customer will announce local routes only.

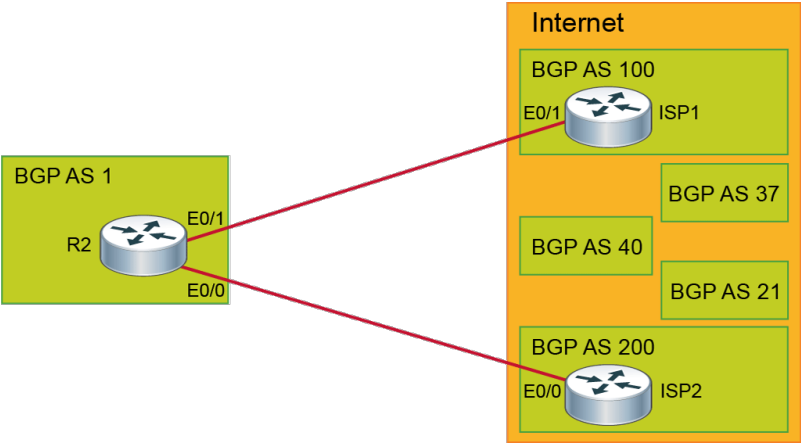
Discovery 7: Configure Non-Transit Autonomous System

Overview

Through this discovery, you will learn how to configure [BGP AS](#) path filters as a route filtering mechanism. You will prevent transmitting routes, that are received on router R2, from one [ISP](#) router to other ISP router. This way, you make sure that customer's autonomous system AS1 does not become transit AS.



Topology



Job Aids

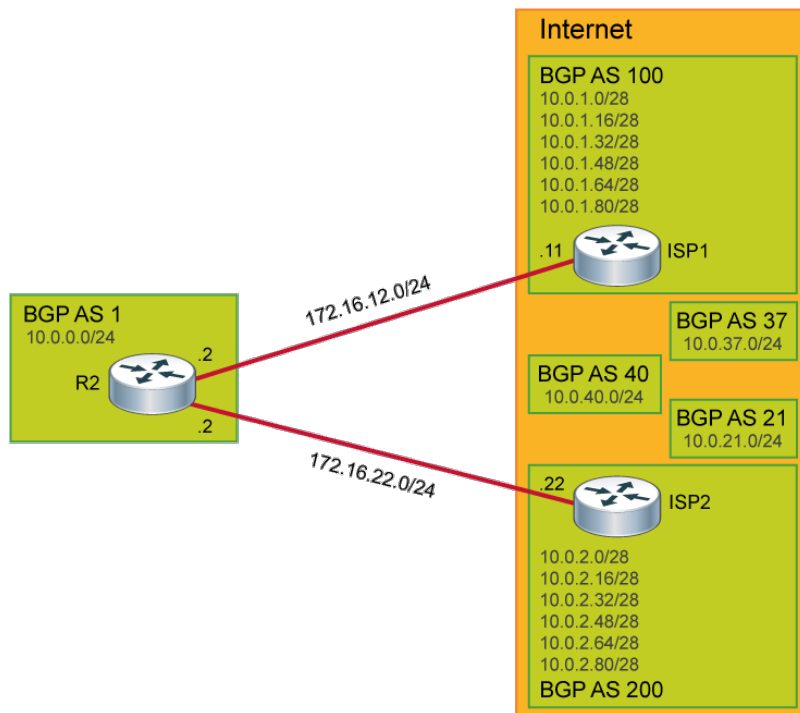


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP1	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate more networks in different autonomous systems.
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks
ISP2	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate more networks in different autonomous systems.

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#) (R2 to ISP1 and R2 to ISP2).

Configure AS Path Filters

Discovery Steps

Step 1

On the ISP2 router, verify that you receive routes that the router ISP1 advertises.

```
ISP2# show ip bgp
BGP table version is 32, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/24       172.16.22.2          0             0 1 i
*> 10.0.1.0/28       172.16.22.2          0             0 1 100 i
*> 10.0.1.16/28      172.16.22.2          0             0 1 100 i
*> 10.0.1.32/28      172.16.22.2          0             0 1 100 i
*> 10.0.1.48/28      172.16.22.2          0             0 1 100 i
*> 10.0.1.64/28      172.16.22.2          0             0 1 100 i
*> 10.0.1.80/28      172.16.22.2          0             0 1 100 i
*> 10.0.2.0/28       0.0.0.0              0            32768 i
*> 10.0.2.16/28      0.0.0.0              0            32768 i
*> 10.0.2.32/28      0.0.0.0              0            32768 i
*> 10.0.2.48/28      0.0.0.0              0            32768 i
*> 10.0.2.64/28      0.0.0.0              0            32768 i
*> 10.0.2.80/28      0.0.0.0              0            32768 i
*> 10.0.2.96/28      0.0.0.0              0            32768 i
   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.2.112/28     0.0.0.0              0            32768 i
*> 10.0.2.128/28     0.0.0.0              0            32768 i
*> 10.0.21.0/24      0.0.0.0              0            32768 i
* 10.0.37.0/24       172.16.22.2          0             0 1 100 37 i
*> 0.0.0.0           0.0.0.0              0            32768 i
* 10.0.40.0/24       172.16.22.2          0             0 1 100 37 40 i
*> 0.0.0.0           0.0.0.0              0            32768 i
```

You can see that ISP1 router in AS 100 either originates or advertises all highlighted routes to the router R2:

- Networks 10.0.1.0/28, 10.0.1.16/28, 10.0.1.32/28, 10.0.1.48/28, 10.0.1.64/28 and 10.0.1.80/28 were originated by ISP1 router in AS 100 and advertised to the router R2.
- Router ISP 1 advertises networks 10.0.37.0/24 and 10.0.40.0/24 to the router R2.
- The customer router R2 originates route 10.0.0.0/24.

All these routes have next hop set to IP address 172.16.22.2, which is the IP address of the R2 router.

Step 2

On the R2 router, verify which routes are being advertised to EBGp neighbor router ISP2.

```
R2# show ip bgp neighbors 172.16.22.22 advertised-routes
BGP table version is 22, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/24       0.0.0.0              0            32768 i
*> 10.0.1.0/28       172.16.12.11          0             0 100 i
*> 10.0.1.16/28      172.16.12.11          0             0 100 i
*> 10.0.1.32/28      172.16.12.11          0             0 100 i
*> 10.0.1.48/28      172.16.12.11          0             0 100 i
*> 10.0.1.64/28      172.16.12.11          0             0 100 i
*> 10.0.1.80/28      172.16.12.11          0             0 100 i
*> 10.0.2.0/28       172.16.22.22          0             0 200 i
*> 10.0.2.16/28      172.16.22.22          0             0 200 i
*> 10.0.2.32/28      172.16.22.22          0             0 200 i
*> 10.0.2.48/28      172.16.22.22          0             0 200 i
*> 10.0.2.64/28      172.16.22.22          0             0 200 i
*> 10.0.2.80/28      172.16.22.22          0             0 200 i
```

```

*> 10.0.2.96/28      172.16.22.22      0      0 200 i
Network      Next Hop      Metric LocPrf Weight Path
*> 10.0.2.112/28     172.16.22.22      0      0 200 i
*> 10.0.2.128/28     172.16.22.22      0      0 200 i
*> 10.0.21.0/24      172.16.22.22      0      0 200 21 i
*> 10.0.37.0/24      172.16.12.11      0      0 100 37 i
*> 10.0.40.0/24      172.16.12.11      0      0 100 37 40 i

Total number of prefixes 19

```

You can see that ISP1 router in AS 100 either originates or advertises all highlighted routes to the router R2:

- The customer router R2 advertises the locally originated route 10.0.0.0/24 to the router ISP2.
- The customer router R2 advertises networks 10.0.1.0/28, 10.0.1.16/28, 10.0.1.32/28, 10.0.1.48/28, 10.0.1.64/28 and 10.0.1.80/28 to the router ISP2.
- The customer router R2 advertises networks 10.0.37.0/24 and 10.0.40.0/24 to the router ISP2.

Step 3

Configure AS path filtering on the R2 router, so that only locally originated network will be advertised to both ISP1 and ISP2 routers.

```

R2(config)# ip as-path access-list 1 permit ^$
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 filter-list 1 out
R2(config-router)# neighbor 172.16.22.22 filter-list 1 out

```

Step 4

Clear all BGP sessions on the R2 router.

```

R2# clear ip bgp *

```

Step 5

On the ISP2 router, verify that you do not receive routes that the ISP1 router advertises.

```

ISP2# show ip bgp
BGP table version is 26, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network      Next Hop      Metric LocPrf Weight Path
*> 10.0.0.0/24    172.16.22.2    0      0 1 i
*> 10.0.2.0/28    0.0.0.0        0      32768 i
*> 10.0.2.16/28   0.0.0.0        0      32768 i
*> 10.0.2.32/28   0.0.0.0        0      32768 i
*> 10.0.2.48/28   0.0.0.0        0      32768 i
*> 10.0.2.64/28   0.0.0.0        0      32768 i
*> 10.0.2.80/28   0.0.0.0        0      32768 i
*> 10.0.2.96/28   0.0.0.0        0      32768 i
*> 10.0.2.112/28  0.0.0.0        0      32768 i
*> 10.0.2.128/28  0.0.0.0        0      32768 i
*> 10.0.21.0/24   0.0.0.0        0      32768 i
*> 10.0.37.0/24   0.0.0.0        0      32768 i
*> 10.0.40.0/24   0.0.0.0        0      32768 i

```

The only route that is received from the router R2 with next hop IP address 172.16.22.2 is 10.0.0.0/24, which is locally originated network on customer's router R2.

Step 6

On the R2 router, verify which routes are being advertised to EBGp neighbor router ISP2.

```

R2# show ip bgp neighbors 172.16.22.22 advertised-routes
BGP table version is 22, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,

```

```

      r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
      x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network      Next Hop      Metric LocPrf Weight Path
*> 10.0.0.0/24    0.0.0.0              0         32768 i

Total number of prefixes 1
```

You should only see the network 10.0.0.0/24 being advertised to the EBGP neighbor ISP2. This network is locally originated on the R2 router. You made the customer AS1 a non-transit autonomous system.

Monitoring AS Path Filters

The Cisco IOS commands that are most frequently used to monitor the operation of configured [AS](#) path filters include **show ip as-path-access-list**, **show ip bgp regexp**, and **show ip bgp filter-list**.

```
router# show ip as-path-access-list [filter list]
```

- Displays all routes in the BGP table matching regular-expression in one or all filter-lists

```
router# show ip bgp regexp regular-expression
```

- Displays one or all filter-lists

```
router# show ip bgp filter-list access-list-number
```

- Displays all routes in the BGP table that the specified AS path access-list permits

show ip as-path-access-list Command

- Displaying configured filters:

```
R2# show ip as-path-access-list
AS path access list 1
    permit ^$
AS path access list 7
    deny _21_
    permit .*
AS path access list 25
    permit 40$
```

The **show ip as-path-access-list** command displays a specific access-list or all AS path access-lists in the router.

show ip bgp regexp Command

- Routes that an expression matches:

```
R2# show ip bgp regexp _21_
BGP table version is 20, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.21.0/24      172.16.22.22           0             0 200 21 i
*                   172.16.12.11           0             0 100 37 40 2
1 i
* 10.0.37.0/24       172.16.22.22           0             0 200 21 40 3
7 i
* 10.0.40.0/24       172.16.22.22           0             0 200 21 40 i
```

Because regular expressions sometimes get complex, thorough testing of them is required. The **show ip bgp regexp** command displays all routes currently in the [BGP](#) table that have an AS path attribute that matches the typed-in regular expression. Use the **show ip bgp regexp** command to test a regular expression that is typed in on the command line. The result is a printout on the screen of all those routes currently in the BGP table that had an AS path attribute matching the typed-in regular expression.

In the example, you wish to find all BGP routes passing through AS 21.

show ip bgp regexp Command

To display routes that match an AS path regular expression, use the **show ip bgp regexp** privileged EXEC command:

```
show ip bgp regexp regular-expression
```

Syntax Description

Parameter	Description
<i>regular-expression</i>	Regular expression to match the BGP AS paths

- Routes that a filter-list matches:

```
R2# show ip as-path-access-list 25
AS path access list 25
  permit 40$

R2# show ip bgp filter-list 25
BGP table version is 20, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*   10.0.40.0/24    172.16.22.22             0         0 200 21 40 i
*>  172.16.12.11    172.16.12.11             0         0 100 37 40 i
```

An AS path access-list is even more complex because it is a combination of several regular expressions. There is one expression on each access-list line. Use the **show ip bgp filter-list** command to test the entire AS path access-list. The result is a printout on the screen of all the routes currently in the BGP table that the access-list permits their AS path attribute.

The AS path access-list number 25 in the example figure consists of one single line. It permits the routes that originate in AS 40. The **show ip bgp filter-list** command displays all the routes currently in the BGP table that the AS path access-list 25 permits.

show ip bgp filter-list Command

To display routes that conform to a specified filter-list, use the **show ip bgp filter-list** privileged EXEC command:

```
show ip bgp filter-list access-list-number
```

Syntax Description

Parameter	Description
<i>access-list-number</i>	Integer from 1 to 199 that indicates the regular expression access-list number

Summary

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

- Several scenarios require BGP route-filtering that is based on AS path, including:
 - Announcing only local routes to the ISP (AS path needs to be empty).
 - Selecting routes that are based on a specific AS number in the AS path.
 - Accepting routes for a specific AS only from some BGP neighbors
- By applying specific selection criteria to the contents of the AS path attribute, routers can select a subset of routes from the total set of routes that are received.
- Cisco IOS software internally translates the AS path encoding, which is carried with all BGP routes into a character string. This string is then tested against the regular expression.
- String matching operates when you are using AS path regular expressions to match BGP routes.
- You can use AS path filters to select those routes that will be allowed.
- AS path access-list creates an AS path filter, which is applied to a set of routes from which to select a subset. The **ip as-path access-list** global configuration command defines a BGP AS path access-list, and the **neighbor filter-list** router configuration command sets up a BGP filter.
- There are a number of Cisco IOS commands that are required to monitor the operation of configured AS path filters, including **show ip as-path-access-list**, **show ip bgp regexp**, and **show ip bgp filter-list**.

Filtering with Prefix Lists

Overview

Where multiple paths between a customer and an [ISP](#) exist, there is a requirement to filter certain information during [BGP](#) updates. You filter the information to influence the route selection or to enforce an administrative policy. To meet this requirement, you must use filters. Using prefix lists is typically easier than using standard IP access lists and provides performance benefits. It is important for you to understand the commands to apply filtering of inbound or outbound updates with prefix lists and where they should be applied.

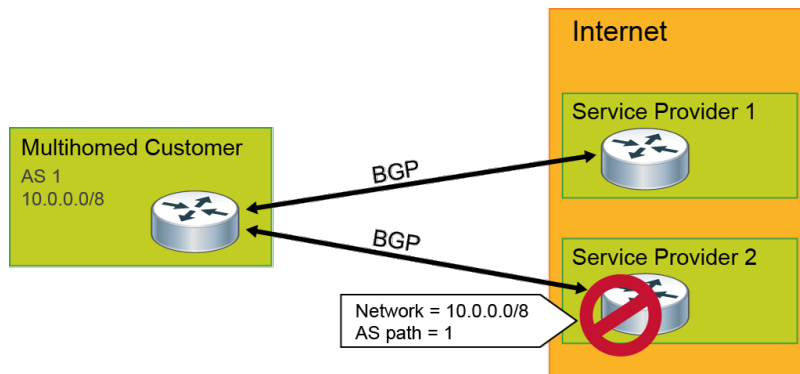
In this lesson, you will learn the requirements for using prefix-based filters in customer implementations where connections to multiple ISPs must be supported. You will also learn the advantages of prefix lists over IP access lists. The commands to apply filtering of inbound or outbound updates with prefix lists and to configure prefix list filters are discussed, and also where network administrators should apply them.

Upon completing this lesson, you will be able to:

- Identify the requirement for prefix-based filters in network implementations where multiple connections between a customer and ISPs exist
- List the advantages of prefix lists versus IP access lists
- Identify the Cisco IOS command that is required to configure prefix list filters
- Describe where you can implement prefix lists in a BGP network
- Identify the Cisco IOS commands that are required to apply filtering of inbound or outbound updates with prefix lists
- Identify the Cisco IOS commands that are required to modify configured prefix-list filters
- Identify the Cisco IOS commands that are required to monitor the operation of configured prefix list filters

Requirements for Prefix-Based Filters

Customers with multihomed networks are responsible for announcing their own networks using [BGP](#). Typically, customers are not as experienced with BGP as service providers, and therefore problems are more likely to occur. A service provider with a multihomed customer must take precautions not to accept, use, or forward any erroneous routing information that is received from the customer.



- Service providers have to filter customer updates to ensure that the customers announce only their assigned address space.

The customer is assigned a set of IP network numbers that it should announce. If the customer announces any additional networks, something is wrong. The customer may have forgotten not to act as a transit [AS](#) and may have started propagating routes that it has received from the other service provider. Or, the customer may have accidentally started to announce its private address space. The customer may use this private address space for address links, loopback interfaces, or other devices that should never access the Internet.

To avoid problems, the service provider can apply an IP prefix filter on the incoming information from the customer. The service provider will accept only network numbers that the access list or prefix list permits.

Prefix Lists vs. IP Access Lists

Traditionally, the filtering of IP network numbers has been accomplished using an access list. The access list is then bound to either the incoming or outgoing information of a neighbor by using the **neighbor distribute-list** command. A [BGP](#) update about a network number that the access list permits will be accepted. An update about a network number that the access list denies will be dropped.

Traditional prefix filters

- Traditional IP prefix filters were implemented with IP access lists configured with the **distribute-list** command.
- IP access lists used as route filters have several drawbacks:
 - Subnet mask cannot be easily matched.
 - Access lists are evaluated sequentially for every IP prefix in the routing update.
 - Extended access lists can be cumbersome to configure.

However, standard access lists do not support the testing of the subnet masks. If the access list permits 10.0.0.0/16, it would also permit 10.0.0.0/8.

Extended access lists can do testing on both an IP network number and a subnet mask, but the syntax is cumbersome.

Prefix lists

- Route-filtering mechanism
- Significant performance improvement on long filters
 - Inside Cisco IOS software, the prefix list is a tree structure and is not scanned sequentially.
- Support for incremental updates
 - Individual entries in prefix lists can be inserted or deleted.
- More user-friendly CLI
 - The CLI for using access lists to filter BGP updates is difficult to understand and use, because it uses the packet-filtering format.
- Greater flexibility; can match on subnet masks

The **ip prefix-list** configuration command has several benefits if you compare it to the **access-list** command. The intended use of prefix lists is limited to route filtering, whereas access lists were initially intended for packet filtering, which was then extended to filter routes.

The prefix list is internally transformed into a tree structure, with each branching of the tree serving as a test. Cisco IOS software determines a verdict of either "permit" or "deny" much faster this way, compared to sequentially interpreting an access list.

The configuration [CLI](#) that you use when configuring the **ip prefix-list** command, allows you to assign a line number to each line of the prefix list. The router uses this number to sort the entries in the prefix list. If the lines are initially assigned line numbers, with some spacing in between them, administrators can later insert extra lines. Individual lines can also be removed without removing the entire list.

Routers match network numbers in a routing update against the prefix list, using as many bits as indicated. For example, a prefix list can be specified to be 10.0.0.0/16, which will match 10.0.0.0 routes but not 10.1.0.0 routes.

Optionally, the prefix list can also specify the size of the subnet mask. In addition, the prefix list can indicate that the subnet mask must be in a specified range.

- Key access list features are preserved.
 - Filtering using "permit" or "deny"
 - Order dependency (first match wins)
 - Security-focused: no match means "deny"
- The matching mechanism has changed.
 - Matches routes in a part of address space with subnet mask longer or shorter than a set number

The prefix list shares several similarities with the access list. It can consist of any number of lines, each of which indicates a test and a result. The router can interpret the lines in the specified order, although this process is optimized in the Cisco IOS software. When a router evaluates a route against the prefix list, the first line that

matches results in either a "permit" or "deny." If none of the lines in the list match, the result is "implicitly deny."

Testing is done using prefixes. The indicated number of bits in the prefix is compared with the same number of bits in the network number in the update. If the bits match, testing continues with an examination of the number of bits set in the subnet mask. The prefix list line can indicate a range within which the number must fall to pass the test. If no range is indicated, the subnet mask must match the prefix size.

Configuring Prefix Lists

You can filter certain information during [BGP](#) updates using prefix list filters to influence route selection or to enforce an administrative policy.

```
router(config)# ip prefix-list list-name [seq seq] {permit|deny} network/len [ge value] [le value]
```

- Prefix lists have names and sequence numbers (like route-maps).
- An entry with no **le** or **ge** parameter matches exactly the specified prefix.
- An entry with an **le** or **ge** parameter matches any route within the address space of *address/prefix*. But the prefix must be longer or equal to **ge value** and shorter than or equal to **le value**.

ip prefix-list Configuration Command

To create an entry in a prefix list, use the **ip prefix-list global** configuration command.

```
ip prefix-list list-name [seq seq-value] {permit | deny} network/len [ge ge-value] [le le-value]
```

Syntax Description

Parameter	Description
<i>list-name</i>	Name of a prefix
seq	(Optional) Applies the sequence number to the prefix
<i>seq-value</i>	(Optional) Specifies the sequence number for the prefix
permit	Permits access for matching conditions
deny	Denies access to matching conditions
<i>network/len</i>	(Mandatory) The network number and length (in bits) of the subnet mask
ge	(Optional) Applies the <i>ge-value</i> to the range specified
<i>ge-value</i>	(Optional) Specifies the lesser value of a range (the "from" portion of the range description)
le	(Optional) Applies the <i>le-value</i> to the range specified
<i>le-value</i>	(Optional) Specifies the greater value of a range (the "to" portion of the range description)

When multiple entries of a prefix list match a given prefix, the sequence number of a prefix-list entry identifies the entry with the lowest sequence number. In this case, the entry with the smallest sequence number is considered to be the "real" match.



You can specify sequence values for prefix list entries in any increments that you want (the automatically generated numbers are increased in units of 5). If you specify the sequence values in increments of 1, you will not be able to insert more entries into the prefix list. If you choose very large increments, you could run out of sequence values.

You can use the parameters **ge** and **le** to specify the range of the prefix length to be matched for prefixes that are more specific than *network/len*. The exact match is assumed when neither **ge** nor **le** is specified. The range is assumed to be from *ge-value* to 32 only if the **ge** attribute is specified. The range is assumed to be from *len* to *le-value* only if the **le** attribute is specified.

Prefix list matching rules

- Prefix list entries with no **ge** or **le** option match only the specified route.
 - Similar to IP access lists with no wildcard bits
 - Matching also considers the subnet mask

Which of the following routes will be matched by:

```
ip prefix-list MyList permit 192.168.0.0/16?
```

192.168.0.0/16  192.168.0.0/20  192.168.2.0/16 

Prefix list entries without the **ge** or **le** option match only the route with the specified [IP address](#) and subnet mask. In the example here, the prefix list entry **permit 192.168.0.0/16** will not match the route 192.168.2.0/24 because of the mismatch in the IP address. It will also not match the route 192.168.0.0/20 because of the mismatch in the subnet mask.

- A prefix list entry with **ge** or **le** option matches any prefix within specified address space where the subnet mask falls within the specified limits.

Which of the following routes will be matched by:

```
ip prefix-list MyList permit 192.168.0.0/16 le 20?
```

192.168.0.0/16  192.168.17.0/20  192.168.2.0/24 

```
ip prefix-list MyList permit 192.168.0.0/16 ge 18?
```

192.168.0.0/16  192.168.17.0/20  192.168.2.0/24 

Prefix list entries with the **ge** or **le** option specified match any prefix within the address space that the *network/len* parameter specifies, under one condition. The subnet mask length of the route must fall within the range that the **le** and **ge** parameters specify.

In the first example in the figure, the route 192.168.2.0/24 is not matched by the prefix list entry **permit 192.168.0.0/16** even though the IP address falls within the specified address range. It is because the subnet mask is too long.

In the second example, the route 192.168.0.0/16 is not matched by prefix list entry **permit 192.168.0.0/18** because the subnet mask is too short.

Configuring Prefix Lists Example

What will be matched by:

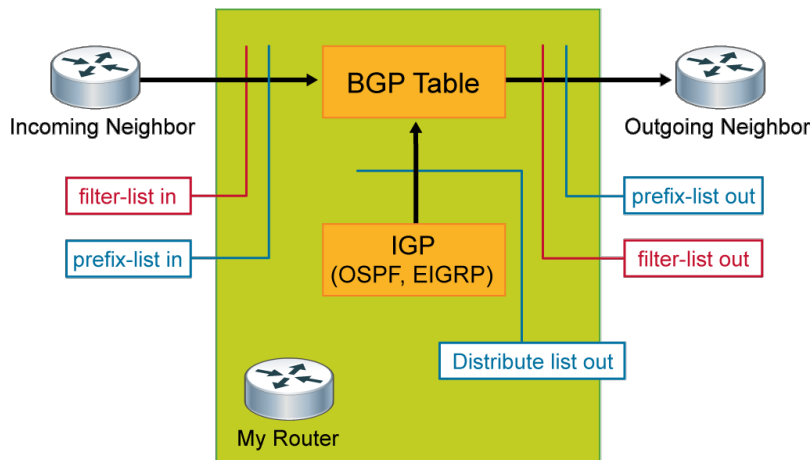
- a) ip prefix-list A **permit 0.0.0.0/0 ge 32**
 - b) ip prefix-list B **permit 128.0.0.0/2 ge 17**
 - c) ip prefix-list C **permit 0.0.0.0/0 le 32**
 - d) ip prefix-list D **permit 0.0.0.0/0**
 - e) ip prefix-list E **permit 0.0.0.0/1 le 24**
- a) All host routes
 - b) Any subnet in class B address space
 - c) All routes
 - d) Just the default route
 - e) Any prefix in class A address space covering at least 256 addresses

The list contains some commonly used prefix list examples.

In the list, prefix list A **permit 0.0.0.0/0 ge 32** will match all host routes. Prefix list B **permit 128.0.0.0/2 ge 17** will match any subnet in a class B address space. Prefix list C **permit 0.0.0.0/0 le 32** will match all routes, but only the default route will be matched by prefix list D **permit 0.0.0.0/0**. Finally, any prefix in a class A address space that covers at least 256 addresses will be matched to prefix list E **permit 0.0.0.0/1 le 24**.

BGP Filters Implementation

You can optionally apply filter lists and prefix lists on either incoming or outgoing neighbors in any combination. Both the incoming prefix list and the incoming filter list must permit the routes that are received from a neighbor before they are accepted into the **BGP** table. Outgoing routes must pass both the outgoing filter list and the outgoing prefix list before being transmitted to the neighbor.



When a router is configured to redistribute routing information from an **IGP** into BGP, the following must be true before a route is injected into the BGP table:

- The routes must successfully pass any prefix list or access list that is applied to the redistribution.

Implementing Prefix Lists in the BGP Process

You can use prefix lists to filter incoming or outgoing [BGP](#) updates to neighbors. You can also use prefix lists to filter routes that are being redistributed into the BGP process from other routing protocols.

```
router(config-router)# neighbor {ip-address|peer-group-name} prefix-list prefix-listname {in|out}
```

- Filters inbound or outbound BGP routing updates for a configured neighbor session

```
router(config-router)# distribute-list prefix-list prefix-list out routing-process
```

- Filters routes redistributed from specified routing process into BGP

neighbor prefix-list Command

To distribute BGP neighbor information as specified in a prefix list, use the **neighbor prefix-list** router configuration command:

```
neighbor {ip-address | peer-group-name} prefix-list prefix-listname {in | out}
```

To remove an entry, use the **no** form of this command:

```
no neighbor {ip-address | peer-group-name} prefix-list prefix-listname {in | out}
```

Syntax Description

Parameter	Description
<i>ip-address</i>	IP address of neighbor
<i>peer-group-name</i>	Name of a BGP peer group
<i>prefix-listname</i>	Name of a prefix list
in	Access list is applied to incoming advertisements to that neighbor
out	Access list is applied to outgoing advertisements from that neighbor



A BGP peer group is a group of BGP neighbors with the same update policies. Route maps, distribute lists, filter lists, and so on usually set update policies. Instead of defining the same policies for each separate neighbor, a peer group name is configured on the router, and these policies are assigned to the peer group.

distribute-list out Command

To suppress networks from being advertised in updates, use the **distribute-list out** router configuration command:

```
distribute-list {access-list-number | name | prefix-list prefix-listname} out [interface-name | routing-process | autonomous-system-number]
```

To disable this function, use the **no** form of this command:

```
no distribute-list {access-list-number | name | prefix-list prefix-listname} out [interface-name | routing-process | autonomous-system-number]
```

Syntax Description

Parameter	Description
<i>access-list-number name</i>	Standard IP access list number or name.

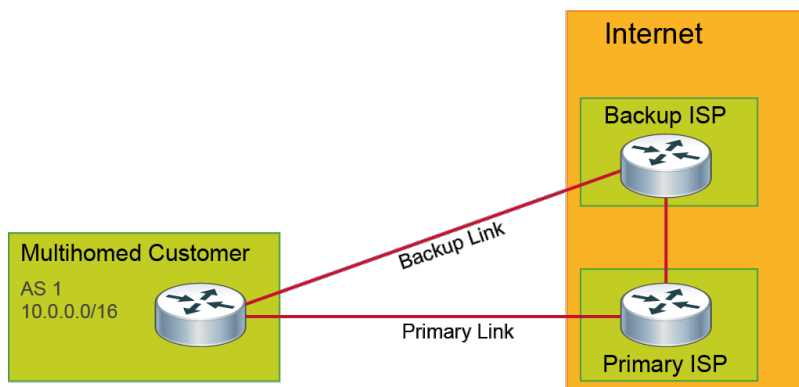
Parameter	Description
	The list defines which networks should be received and which should be suppressed in routing updates.
<i>prefix-listname</i>	Name of a prefix list. The list defines which networks should be received and which should be suppressed in routing updates. This decision is based on matching of the network prefix to the prefixes in the list.
out	Applies the access list to outgoing routing updates.
<i>interface-name</i>	(Optional) Name of a particular interface.
<i>routing-process</i>	(Optional) Name of a particular routing process, or the keyword static or connected .
<i>autonomous-system-number</i>	(Optional) AS number.



Although you can use the **neighbor prefix-list** router configuration command as an alternative to the **neighbor distribute-list** command, do not use both the **neighbor prefix-list** and **neighbor distribute-list** command filtering for the same neighbor in any given direction. These two commands are mutually exclusive, and only one command (**neighbor prefix-list** or **neighbor distribute-list**) can be applied for each inbound or outbound direction.

Prefix List Example: Filtering Customer Prefixes

The figure illustrates filtering customer prefixes.



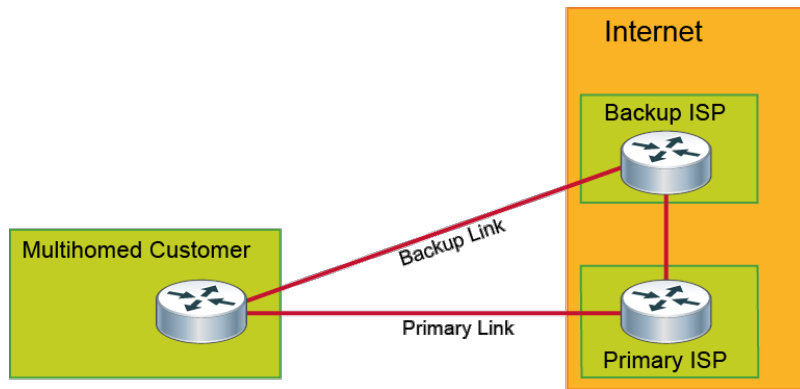
- Requirement: The customer will announce prefixes only from assigned address space (10.0.0.0/16), with subnet masks no longer than /24.

```
router bgp Primary-ISP-AS
neighbor Customer prefix-list Cust-A in
!
ip prefix-list Cust-A permit 10.0.0.0/16 le 24
```

In this example, a multihomed customer has been assigned the address space 10.0.0.0/16. The customer may subnet this address space but may not announce subnets smaller than a subnet mask of 255.255.255.0. Larger subnets are accepted. If the customer has subnetted the network into smaller subnets, it must summarize the routing information about those subnets into at least /24 prefixes before announcing them.

The primary **ISP** implements a prefix list with a Cust-A name, to perform the filtering of incoming information from the multihomed customer. The prefix list permits all routes that are received from the customer that have 10.0 in the first 16 bits and have a subnet mask of 24 bits or less. Any other routes from the customer are denied and silently ignored.

Prefix List Example: Filtering Peer Prefixes



- Requirement: The ISP will not accept routes with subnet masks longer than /24; subnet masks from class B address space will be no longer than /20.

```
router bgp Primary-ISP-AS
neighbor Backup-ISP prefix-list Peer in
!
ip prefix-list Peer seq 5 permit 128.0.0.0/2 le 20
ip prefix-list Peer seq 10 permit 0.0.0.0/0 le 24
```

In this example, the primary ISP will not accept any route from the customer that indicates a subnet smaller than a 255.255.255.0 subnet mask. The class B network, however, must not be subnetted into subnets smaller than a 255.255.240.0 subnet mask.

The primary ISP implements this rule by using a prefix list named Peer. The first line in the prefix list checks whether it is a class B network. Remember that a class B address always has the binary sequence 10 as the first 2 bits in the first byte. The second line matches any prefix.

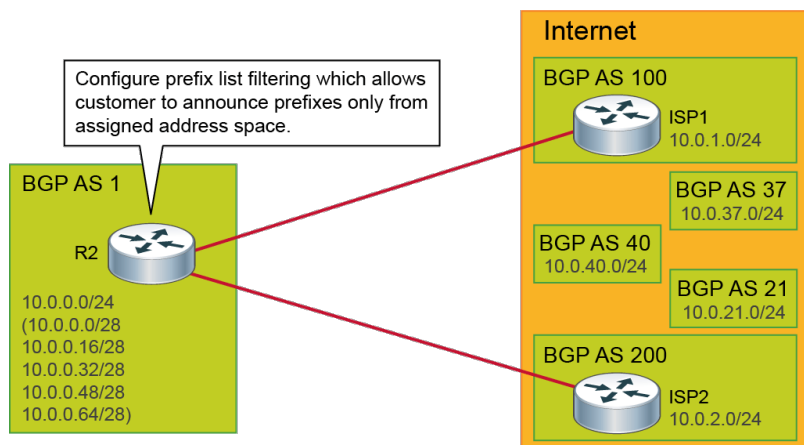
When the primary ISP receives a route from the customer, it compares the route with both lines. If the route is a class B network, both lines match. Testing continues with checking the subnet mask. An upper bound is explicitly indicated, giving a maximum prefix length of 20 bits.

If the received route is not a class B network, only the second line matches. In this case, the subnet mask length must be greater than or equal to 0 and less than or equal to 24. This rule provides a route less explicit than a /24 prefix.

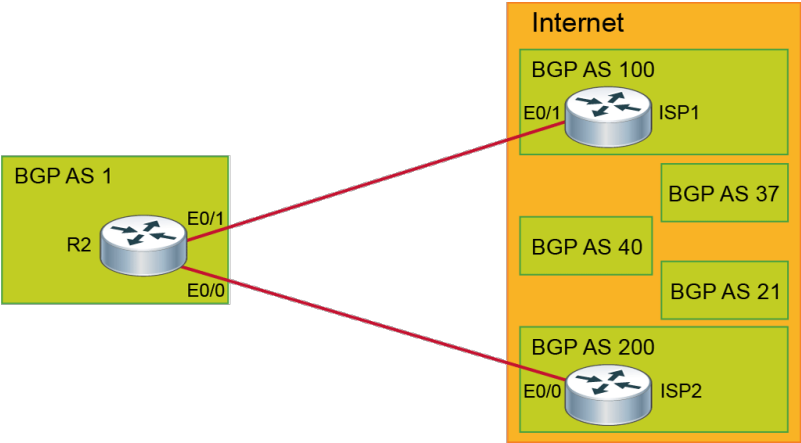
Discovery 8: Filtering Customer Prefixes

Overview

Through this discovery, you will learn how to configure a prefix list to filter customer prefixes. You as an ISP1 service provider will allow the customer on the router R2 to announce prefixes only from the assigned address space (10.0.0.0/24). You will not allow the customer to further subnet the assigned address space, thus subnet mask should be 24.



Topology



Job Aids

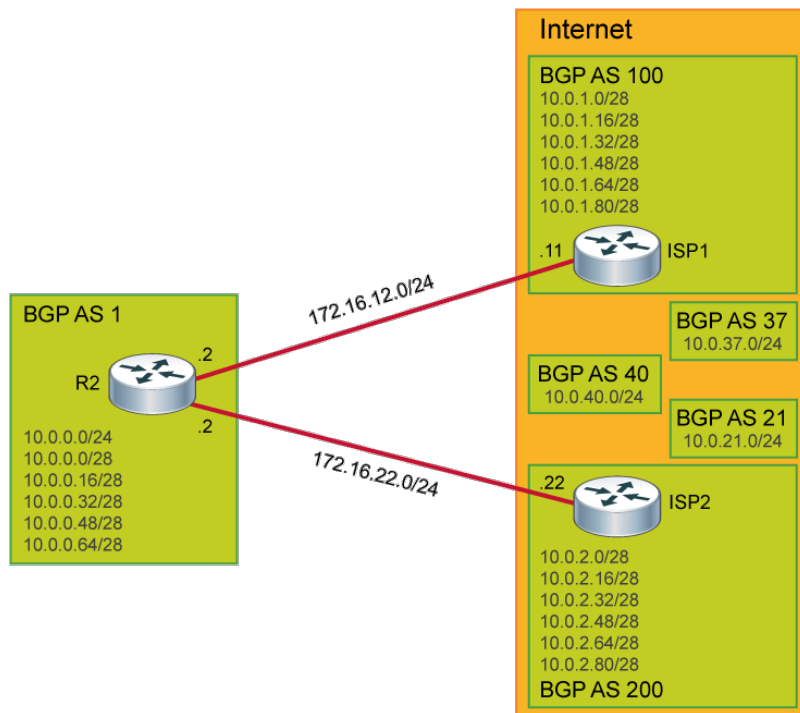


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP1	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate additional networks in different autonomous systems.
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28	Loopbacks simulate LAN networks
ISP2	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate additional networks in different autonomous systems.

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#) (R2 to ISP1 and R2 to ISP2).

The R2 router announces these networks:

- 10.0.0.0/24 (assigned address space to the Customer)
- 10.0.0.16/28
- 10.0.0.32/28
- 10.0.0.48/28
- 10.0.0.64/28

Filtering Customer Prefixes

Discovery Steps

Step 1

On the ISP1 router, verify that you receive routes with mask larger than /24 from the customer router R2.

```
ISP1# show ip bgp
BGP table version is 15, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/24	172.16.12.2	0		0	1 i
*> 10.0.0.16/28	172.16.12.2	0		0	1 i
*> 10.0.0.32/28	172.16.12.2	0		0	1 i
*> 10.0.0.48/28	172.16.12.2	0		0	1 i
*> 10.0.0.64/28	172.16.12.2	0		0	1 i
*> 10.0.1.0/28	0.0.0.0	0		32768	i
*> 10.0.1.16/28	0.0.0.0	0		32768	i
*> 10.0.1.32/28	0.0.0.0	0		32768	i
*> 10.0.1.48/28	0.0.0.0	0		32768	i
*> 10.0.1.64/28	0.0.0.0	0		32768	i
*> 10.0.1.80/28	0.0.0.0	0		32768	i
*> 10.0.21.0/24	0.0.0.0	0		32768	i
*> 10.0.37.0/24	0.0.0.0	0		32768	i
*> 10.0.40.0/24	0.0.0.0	0		32768	i

You can see that customer announces four networks with mask /28.

Step 2

On the R2 router, verify that customer router R2 originates assigned network 10.0.0.0/24 and four subnets from this assigned address space with mask /28.

```
R2# show ip bgp neighbors 172.16.12.11 advertised-routes
BGP table version is 58, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/24	0.0.0.0	0		32768	i
*> 10.0.0.16/28	0.0.0.0	0		32768	i
*> 10.0.0.32/28	0.0.0.0	0		32768	i
*> 10.0.0.48/28	0.0.0.0	0		32768	i
*> 10.0.0.64/28	0.0.0.0	0		32768	i

Total number of prefixes 5

Customer assigned address space is 10.0.0.0/24, which is further subnetted to four /28 subnets, which are also announced in BGP updates.

Step 3

Configure prefix list filtering on the ISP1 router to allow the customer on the R2 router to announce prefixes only from the assigned address space (10.0.0.0/24).

You should not allow the customer to further subnet the assigned address space, thus subnet mask should be 24.

```
ISP1(config)# ip prefix-list Customer permit 10.0.0.0/24
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.12.2 prefix-list Customer in
```

Step 4

Clear all BGP sessions on the R2 router.


```
R2# clear ip bgp *
```

Step 5

On the ISP1 router, verify that you receive routes with mask /24 only from the customer router R2.

```
ISP1# show ip bgp
BGP table version is 21, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/24      172.16.12.2          0         0 1 i
*> 10.0.1.0/28      0.0.0.0              0        32768 i
*> 10.0.1.16/28     0.0.0.0              0        32768 i
*> 10.0.1.32/28     0.0.0.0              0        32768 i
*> 10.0.1.48/28     0.0.0.0              0        32768 i
*> 10.0.1.64/28     0.0.0.0              0        32768 i
*> 10.0.1.80/28     0.0.0.0              0        32768 i
*> 10.0.21.0/24     0.0.0.0              0        32768 i
*> 10.0.37.0/24     0.0.0.0              0        32768 i
*> 10.0.40.0/24     0.0.0.0              0        32768 i
```

You can see that only customer network 10.0.0.0 with mask /24 is put into the BGP table because of the prefix list filtering configured.

Step 6

On the R2 router, verify that customer router R2 still originates assigned network 10.0.0.0/24 and four subnets from this assigned address space with mask /28.

```
R2# show ip bgp neighbors 172.16.12.11 advertised-routes
BGP table version is 58, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/24      0.0.0.0          0         32768 i
*> 10.0.0.16/28     0.0.0.0          0         32768 i
*> 10.0.0.32/28     0.0.0.0          0         32768 i
*> 10.0.0.48/28     0.0.0.0          0         32768 i
*> 10.0.0.64/28     0.0.0.0          0         32768 i

Total number of prefixes 5
```

This way, you confirm that filtering of BGP updates occurs on the ISP1 router.

Modifying Prefix Lists

Lines in a prefix list are assigned sequence numbers. These assignments significantly improve the manageability of the list. These sequence number assignments provide the opportunity to remove a specific line. If the spacing between the sequence numbers allows, they also give you the ability to insert a line between two existing lines.

```
router# show ip prefix-list list-name [detail|summary]
```

- Displays the prefix list and the sequence numbers

```
router(config)# no ip prefix-list seq seq condition
```

- Erases the line with the specified sequence number from the prefix list

```
router(config)# ip prefix-list seq seq condition
```

- Inserts the line into the prefix list at the specified point

To display a currently configured prefix list and its sequence numbers, use the **show ip prefix-list** command with the **detail** keyword.

You can specify sequence values for prefix list entries in any increments that you want (the automatically generated numbers are increased in units of 5). If you specify the sequence values in increments of 1, you will not be able to insert more entries into the prefix list. If you choose very large increments, you could run out of sequence values.

Monitoring Prefix Lists

You can monitor prefix lists by displaying information about a prefix list or by using prefix lists for selective filtering of [BGP](#) table output on Cisco IOS devices.

```
router# show ip prefix-list [detail | summary] prefix-list-name [network/length] [seq sequence-number] [longer] [first-match]
```

- To display information about a prefix list or prefix-list entries

```
router# show ip bgp prefix-list prefix-list-name
```

- Displays all routes in the BGP table matching the prefix list.
- Used for easier monitoring of a desired network prefix group in the BGP table.

show ip prefix-list Command

To display information about a prefix list or prefix list entries, use the **show ip prefix-list EXEC** command:

```
show ip prefix-list [detail | summary] name [network/len] [seq seq-num] [longer] [first-match]
```

Syntax Description

Parameter	Description
detail summary	(Optional) Displays detailed or summarized information about all prefix lists
<i>name</i>	(Optional) The name of a specific prefix list
<i>network/len</i>	(Optional) The network number and length (in bits) of the network mask
seq	(Optional) Applies the sequence number to the prefix-list entry
<i>seq-num</i>	The sequence number of the prefix
longer	Displays all entries of a prefix that are more specific than the given <i>network/len</i>
first-match	Displays the entry of a prefix that matches the given <i>network/len</i>

You can use **show ip bgp prefix-list** command to display selected routes from a BGP routing table based on the contents of a prefix list. Use this command for selective filtering of BGP table output on Cisco IOS devices based on network prefix groups.

To perform prefix list-based BGP table filtering, follow these steps:

1. Configure a prefix list that permits ranges of networks that are meant to be displayed in the BGP table output.
2. Include a reference to a configured prefix list in the **show ip bgp prefix-list** command.



The support for prefix list BGP table filtering was added in Cisco IOS Software Release 12.2(11)T and 12.0(14)ST.

```
ISP1# show ip prefix-list detail InFilter
ip prefix-list InFilter:
  count: 4, range entries: 3, sequences: 5 - 20, refcount: 2
  seq 5 deny 128.0.0.0/2 le 15 (hit count: 0, refcount: 1)
  seq 10 deny 192.0.0.0/3 ge 25 (hit count: 0, refcount: 2)
  seq 15 deny 193.0.0.0/8 ge 21 (hit count: 0, refcount: 1)
  seq 20 permit 0.0.0.0/0 (hit count: 0, refcount: 1)
```

- Displays details about InFilter prefix list

In this example, the **show ip prefix-list** command has been issued with the **detail** keyword. The output of the command displays detailed information about configured prefix lists. The sequence numbers, the prefix list entries, and the number of times that the corresponding prefix matched each entry.

```
ISP1(config)# ip prefix-list MyFilter permit 10.0.1.0/24 ge 28
```

```
ISP1# show ip bgp prefix-list MyFilter
BGP table version is 21, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	10.0.1.0/28	0.0.0.0	0		32768	i
*>	10.0.1.16/28	0.0.0.0	0		32768	i
*>	10.0.1.32/28	0.0.0.0	0		32768	i
*>	10.0.1.48/28	0.0.0.0	0		32768	i
*>	10.0.1.64/28	0.0.0.0	0		32768	i
*>	10.0.1.80/28	0.0.0.0	0		32768	i

- Displays all entries in BGP table with the first three octets equal to 10.0.1 and with length of a subnet greater or equal to 28

This example shows a simple prefix list-based filtering of the BGP table. The prefix list filter permits all networks with the first three octets equal to 10.0.1 and with length of a subnet greater or equal to 28. The **show ip bgp prefix-list** command output displays only the networks that the prefix list filter permits.

Summary

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

- Customers with multihomed networks are responsible for announcing their own networks using BGP. Service providers with multihomed customers must take precautions not to accept, use, or forward any erroneous routing information that is received from their customers.
- Prefix lists have a number of advantages over access lists, including faster "permit" or "deny" determinations and easier CLI editing.
- Prefix lists are configured using the **ip prefix-list** global configuration command.
- Filter lists and prefix lists can be optionally applied on either incoming or outgoing neighbors in any combination.
- Prefix lists can filter incoming or outgoing BGP updates to neighbors and filter routes that are being redistributed into the BGP process from other routing protocols. Use the **neighbor prefix-list router configuration command** to distribute BGP neighbor information as specified in a prefix list.
- Certain Cisco IOS commands (such as the **show ip prefix-list** command) are used to modify configured prefix list filters.
- To display or monitor statistics about a prefix list or prefix list entries, you can use the **show ip prefix-list EXEC** command.

Using Outbound Route Filtering

Overview

You can use [ORF](#) as an extra mechanism to minimize the number of updates that are requested from a neighbor. It will reduce the link bandwidth consumption and [CPU](#) use when a router requests a route refresh. An ORF also allows filtering of information that external networks should not receive (such as [RFC 1918](#) information). Understanding how to monitor outbound route filtering capabilities is also important. A [BGP](#) neighbor that supports specific ORF capabilities will report those capabilities to a monitoring neighbor and can then send a filter of the supported type to the neighbor.

This lesson discusses the function of outbound route filtering in a BGP network. The format and function of ORF messages are discussed, as well as the commands that enable ORF negotiations and the activation of an ORF prefix list. The commands that are used to trigger a route refresh are also detailed. Finally, there is a discussion on how to monitor the operations of a configured ORF in a BGP network.

Upon completing this lesson, you will be able to:

- Describe the function of outbound route filtering in a BGP network
- Compare standard inbound filtering and outbound route filtering
- Describe the function of BGP prefix-based outbound route filtering
- Describe the format and function of an ORF message
- Identify the Cisco IOS command that is required to enable ORF negotiations and activate an ORF prefix list
- Identify the Cisco IOS command that is used to trigger a route refresh message
- Identify the Cisco IOS command that is required to monitor the operation of a configured ORF

Outbound Route Filtering

Outbound route filtering is a prefix-based [BGP](#) feature that is enabled through the advertisement of [ORF](#) capabilities to peer routers and is integrated in Cisco IOS Software Release 12.2(4)T. The advertisement of the ORF capability indicates that a BGP-speaking router will accept a prefix list from a neighbor. It will apply the prefix list to locally configured ORFs (if any exist). When this capability is enabled, the BGP speaker can install an inbound prefix list filter to the remote peer as an outbound filter, which reduces unwanted routing updates.

- The purpose of outbound route filtering is to reduce the amount of BGP traffic and CPU use needed to process routing updates.
- Routers exchange inbound filter configurations, which are used as outbound filters on neighboring routers.
- Filters are described in ORF entries.
- ORF entries are part of the route refresh message.

The standard route refresh message contains the [AFI](#) for which the refresh is needed. Outbound route filtering is an extra mechanism that is used to minimize the number of updates that are requested from a neighbor.

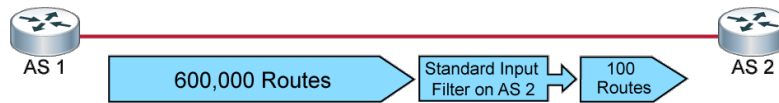
This mechanism reduces the link bandwidth consumption and [CPU](#) use when a router requests a route refresh. Filters that the routers should use with the route refresh are described in ORF entries that are part of the route refresh message.

You can configure the ORF feature with send, receive, or send and receive capabilities. The local peer advertises the ORF capability in the send mode. The remote peer receives the ORF capability in receive mode and applies the filter as the outbound policy. The local and remote peers exchange updates to maintain the ORF for each router. Peer routers exchange updates depending on the ORF prefix list capability that is advertised. The remote peer starts sending updates to the local peer after it receives a route refresh request or an ORF prefix list with an immediate status.

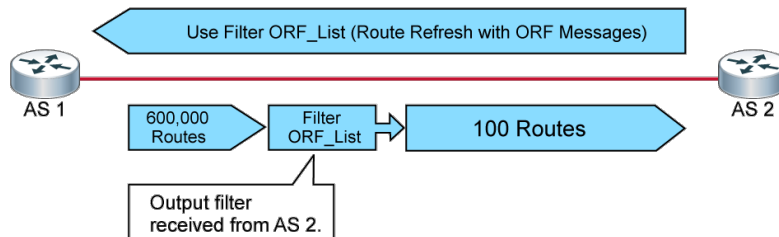
Inbound vs. Outbound Filtering Example

The figure illustrates the comparison between standard inbound filtering and outbound route filtering.

Standard Inbound Filtering:



Outbound Route Filtering:



Outbound route filtering can limit the number of unwanted routing updates, which will reduce the amount of resources that are required for routing update generation and processing. This feature also reduces the amount of resources that are required to receive and discard routes. The receiving router would otherwise filter out these routes, if the [ORF](#) feature were not available.

The example shows two scenarios:

- The first example shows that 600,000 routes are sent from AS1 to AS2 neighbor, and the input filter on AS2 permits only 100 of these routes.
- The second example shows how a route refresh with a filter is sent from AS2 to the AS1 neighbor. The AS1 neighbor then uses the filter before sending the updates. This way, only 100 updates are sent from AS1 to AS2.

BGP Prefix-Based Outbound Route Filtering

The BGP prefix-based outbound route filtering feature uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers. Configuring this feature can help reduce the amount of system resources that are required for generating and processing routing updates by filtering out unwanted routing updates at the source.

- Uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers
- Helps to reduce the amount of system resources that are required for generating and processing routing updates by filtering out unwanted routing updates at the source
- Limits the number of unwanted routing updates, which will reduce the amount of resources that are required for routing update generation and processing
- Reduces the amount of resources that are queried to receive and discard routes that would otherwise be filtered out

The BGP prefix-based outbound route filtering feature is enabled through the advertisement of ORF capabilities to peer routers. The advertisement of the ORF capability indicates that a BGP speaker will accept a prefix list from a neighbor. It will apply the prefix list to locally configured ORFs (if any exist). When this capability is enabled, the BGP speaker can install the inbound prefix list filter to the remote peer as an outbound filter, which reduces unwanted routing updates.

The BGP prefix-based outbound route filtering feature can be configured with send, receive, or send and receive ORF capabilities. The local peer advertises the ORF capability in send mode. The remote peer receives the ORF capability in receive mode and applies the filter as an outbound policy. The local and remote peers exchange updates to maintain the ORF on each router. Updates are exchanged between peer routers by address family depending on the ORF prefix list capability that is advertised. The remote peer starts sending updates to the local peer after a route refresh has been configured with the **clear ip bgp** command. Or after an ORF prefix list with immediate status is processed. The BGP speaker will continue to apply the inbound prefix list to received updates after the speaker pushes the inbound prefix list to the remote peer.

Outbound Route Filter Message

It is important to understand the fields [ORF](#) message.

ORF format

- An ORF message consists of the following fields:
 - AFI/SAFI
 - ORF type
 - When to refresh
 - List of ORF entries
- ORF entries depend on the ORF type.
- The ORF capability needs to be negotiated for every supported ORF type.

An ORF message contains the following information:

- [AFI](#) and [SAFI](#), for which the filter should be used.
- ORF type, which identifies the type of filter.
- When to refresh (immediate or deferred refresh).
- List of ORF entries where the actual filter is defined.

You can use the AFI/SAFI component of the ORF message to provide a coarse level of granular control by limiting the ORF to only the routes whose [NLRI](#) matches the configured AFI/SAFI component.

The router has to negotiate the ORF capability for each ORF type that is supported in the ORF message.

ORF types:

- NLRI (ORF type = 1)
 - Filters that are based on the prefix
- Communities (ORF type = 2)
 - Filters that are based on standard BGP community attributes
- Extended communities (ORF type = 3)
 - Filters that are based on extended BGP community attributes
- Prefix list (ORF type = 128)
 - Filters that are based on Cisco implementation of prefix filtering

The value that is contained in the ORF type determines the content that is contained in the ORF message.

Currently, ORF type 0 is reserved, ORF types 1 to 127 are assigned by the [IANA](#), and ORF types 128 to 255 are vendor-specific (and not assigned by the IANA).

Commonly used ORF types are as follows:

- ORF type 1 is used to filter based on the NLRI.
- ORF type 2 is used to filter based on standard [BGP](#) community attributes.
- ORF type 3 is used to filter based on extended BGP community attributes.
- ORF type 128 is used to filter based on the Cisco proprietary implementation of prefix filtering (prefix lists).

AFI/SAFI is IPv4 unicast.

ORF type is NLRI:

- **Action:** ADD, DELETE, or DELETE ALL
- **Match:** PERMIT or DENY
- **Scope:** EXACT or REFINE
- **NLRI:** Prefix
- **When:** IMMEDIATE or DEFER

The ORF-type setting determines the content of the ORF value. An ORF type of NLRI-based filtering (type 1) uses the following actions:

- **ADD:** Adds a line to a prefix list filter on the remote peer
- **DELETE:** Removes a line from a filter that was previously installed on a remote peer
- **DELETE ALL:** Removes all previously installed filters on the remote peer

For each filter entry, there is a match component that specifies either PERMIT or

DENY. A PERMIT asks the peer to send updates with routes that match the set of entries as specified in the ORF. DENY specifies that the remote peer should not send updates for the entries matching the entries that are specified in the ORF.

For prefixes specified with a match component of PERMIT, the remote peer is asked to pass a prefix with a scope of EXACT (an exact match) or REFINE (its subnets).

Also contained within the ORF message is the when-to-refresh field. A router can set this field to IMMEDIATE—asking the remote peer to refresh as soon as it has finished processing the ORF message. Or it can set it to DEFER—asking the remote peer to wait until it receives a subsequent route refresh message with the same AFI/SAFI.

Configuring Outbound Route Filtering

You need to enable [ORF](#) negotiations and activate an ORF prefix list when configuring outbound route filtering.

```
router(config-router)# neighbor ip-address capability orf prefix-list [receive|send|both]
```

- This command enables negotiation of prefix list ORF capability during session setup.
- The ORF-capable BGP speaker will install ORFs per neighbor.
- **Option:**
 - "Both" allows sending and receiving of prefix lists.
 - "Send" allows only sending of prefix lists.
 - "Receive" allows only receiving of prefix lists.

Cisco routers support the uploading of their prefix lists to a neighbor. You need to use the **neighbor ip-address capability orf prefix-list receive** command to advertise this capability. You need to use **neighbor ip-address capability orf prefix-list send** command to upload the inbound prefix filter to the neighbor. The uploaded filter is then used on the neighboring router after a statically configured outbound prefix list (if it exists) is applied.

The **neighbor ip-address capability orf prefix-list** command enables the negotiation of the prefix-list ORF capability during [BGP](#) session setup. The prefix-list-based ORF (ORF type = 128) is the only ORF type that Cisco IOS software supports.

neighbor orf prefix-list Command

To advertise ORF capabilities to a peer router, use the **neighbor orf prefix-list** command in address family or router configuration mode:

```
neighbor {ip-address} [capability] orf prefix-list [receive | send | both]
```

To disable ORF capabilities, use the **no** form of this command.

```
no neighbor {ip-address} [capability] orf prefix-list [receive | send | both]
```

Syntax Description

Parameter	Description
<i>ip-address</i>	The IP address of the neighbor router
capability	(Optional) Informs the specified neighbor that this router has ORF capabilities
receive	(Optional) Enables the ORF prefix list capability in receive mode
send	(Optional) Enables the ORF prefix list capability in send mode
both	(Optional) Enables the ORF prefix list capability in both receive and send modes



```
AS1:
router bgp 100
 neighbor 172.16.12.2 capability orf prefix-list receive
AS2:
ip prefix-list FILTER seq 5 permit 10.0.1.16/28
!
router bgp 1
 neighbor 172.16.12.11 capability orf prefix-list send
```

```
neighbor 172.16.12.11 prefix-list FILTER in
!
```

- The command **capability orf prefix-list send** on one router requires **capability orf prefix-list receive** on a neighboring router.

The example shows the configuration of two routers where one router has uploaded an input prefix list to the neighbor to be used as an output filter.

The following configuration steps are necessary to enable outbound route filtering:

1. Enable negotiation of outbound filtering based on prefix lists.
2. Attach an input prefix list to a neighbor.
3. Enable sending of input prefix list to the neighbor.

In the example, an ORF has been configured on AS2 to advertise the filter to AS1. An IP prefix list with the name FILTER is created to specify the 10.0.1.16/28 subnet for outbound route filtering. The ORF send capability is configured on AS2 so that AS2 can advertise the ORF to AS1.

AS1 is configured to advertise the ORF receive capability to AS2. AS1 will install the ORF, defined in the FILTER prefix list, after the ORF capabilities have been exchanged. An inbound soft reset needs to be initiated on AS1 at the end of this configuration to activate the ORF.

Using Outbound Route Filtering

Identify the Cisco IOS command that is used to trigger a route refresh message.

```
router# clear ip bgp neighbor in [prefix-filter]
```

- This command triggers a route refresh message.
- This command includes a prefix list in the route refresh message if configured and supported on both ends.
- The prefix list is sent at session setup.
- Use the **prefix-filter** option to refresh the remote filter.

Use the **clear ip bgp neighbor** command with the **prefix-filter** keyword to push out the existing [ORF](#) prefix list so that a new route refresh will be received from a neighbor. The neighbor will use the ORF prefix list that was previously negotiated.

You need to use the **clear ip bgp neighbor** command only when the filter has been modified because the neighbor will store the filter for subsequent route refresh requests. The neighbor will then use the filter on all updates toward the router that originated the filter.



You should enter the **in** keyword when you are using the **clear ip bgp neighbor** command because inbound route refresh is desired; only the inbound prefix list filter is pushed to the neighbor and used by the neighbor in the outbound direction.

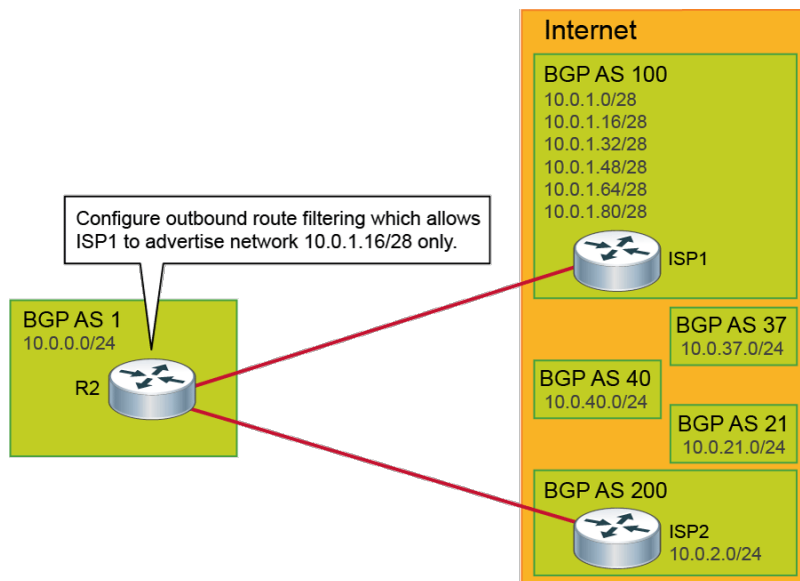
The router will ignore the **prefix-filter** keyword if ORF capability has not been received, or the send capability has not been enabled.

When the **clear ip bgp neighbor** command is used without the **prefix-filter** keyword, a normal route refresh is performed. You should always use the **prefix-filter** keyword when ORF inbound routing policy changes occur.

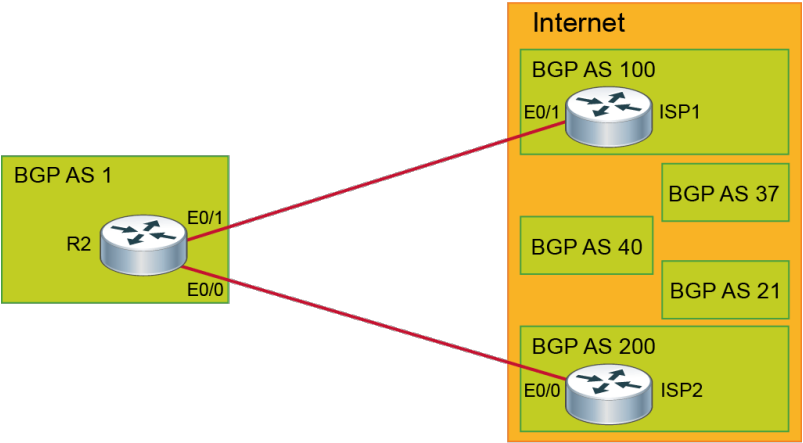
Discovery 9: Prefix-Based Outbound Route Filtering

Overview

Through this discovery, you will learn how to configure prefix-based outbound route filtering. You will create an outbound route filter and configure the R2 router to advertise the filter to the ISP1 router. The filter will allow the ISP1 router to advertise 10.0.1.16/28 network only to the R2 router.



Topology



Job Aids

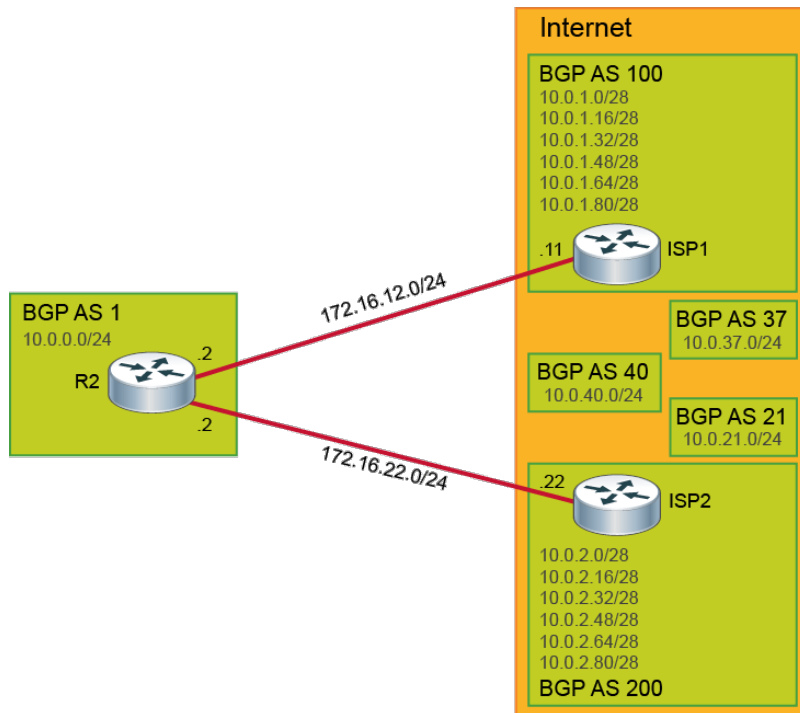


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP1	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate extra networks in different autonomous systems.
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks
ISP2	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate extra networks in different autonomous systems.

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#) (R2 to ISP1 and R2 to ISP2). The ISP1 router announces these networks:

- 10.0.1.0/28 (assigned address space to the Customer)
- 10.0.1.16/28
- 10.0.1.32/28
- 10.0.1.48/28
- 10.0.1.64/28
- 10.0.1.80/28

Prefix-Based Outbound Route Filtering

Discovery Steps

Step 1

On the ISP1 router, verify which networks are advertised via BGP to the R2 router.

```
ISP1# show ip bgp neighbor 172.16.12.2 advertised-routes
BGP table version is 15, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.1.0/28       0.0.0.0            0         32768 i
*> 10.0.1.16/28      0.0.0.0            0         32768 i
*> 10.0.1.32/28      0.0.0.0            0         32768 i
*> 10.0.1.48/28      0.0.0.0            0         32768 i
*> 10.0.1.64/28      0.0.0.0            0         32768 i
*> 10.0.1.80/28      0.0.0.0            0         32768 i
*> 10.0.21.0/24      0.0.0.0            0         32768 i
*> 10.0.37.0/24      0.0.0.0            0         32768 i
*> 10.0.40.0/24      0.0.0.0            0         32768 i
```

You should see these BGP routes being advertised to the R2 router:

- Any subnets in 10.0.1.0/24 range.
- Networks 10.0.21.0/24, 10.0.37.0/24 and 10.0.40.0/24.

Step 2

On the R2 router, verify which routes are received via BGP from the ISP1 router.

```
R2# show ip bgp
BGP table version is 162, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/24       0.0.0.0            0         32768 i
*> 10.0.0.16/28      0.0.0.0            0         32768 i
*> 10.0.0.32/28      0.0.0.0            0         32768 i
*> 10.0.0.48/28      0.0.0.0            0         32768 i
*> 10.0.0.64/28      0.0.0.0            0         32768 i
*> 10.0.1.0/28       172.16.12.11       0           0 100 i
*> 10.0.1.16/28      172.16.12.11       0           0 100 i
*> 10.0.1.32/28      172.16.12.11       0           0 100 i
*> 10.0.1.48/28      172.16.12.11       0           0 100 i
*> 10.0.1.64/28      172.16.12.11       0           0 100 i
*> 10.0.1.80/28      172.16.12.11       0           0 100 i
*> 10.0.2.0/28       172.16.22.22       0           0 200 i
*> 10.0.2.16/28      172.16.22.22       0           0 200 i
*> 10.0.2.32/28      172.16.22.22       0           0 200 i
   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.2.48/28      172.16.22.22       0           0 200 i
*> 10.0.2.64/28      172.16.22.22       0           0 200 i
*> 10.0.2.80/28      172.16.22.22       0           0 200 i
*> 10.0.2.96/28      172.16.22.22       0           0 200 i
*> 10.0.2.112/28     172.16.22.22       0           0 200 i
*> 10.0.2.128/28     172.16.22.22       0           0 200 i
* 10.0.21.0/24       172.16.12.11       0           0 100 37 40 21 i
*>                   172.16.22.22       0           0 200 21 i
*> 10.0.37.0/24      172.16.12.11       0           0 100 37 i
*                   172.16.22.22       0           0 200 21 40 37 i
*> 10.0.40.0/24      172.16.12.11       0           0 100 37 40 i
*                   172.16.22.22       0           0 200 21 40
```

You should see these BGP routes being received on the R2 router from the ISP1 router:

- Any subnets in 10.0.1.0/24 range.
- Networks 10.0.21.0/24, 10.0.37.0/24 and 10.0.40.0/24.

Step 3

Configure the ISP1 router to advertise [ORF](#) receive capability from the R2 neighbor.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.12.2 capability orf prefix-
list receive
```

Step 4

Configure the R2 router to advertise ORF send capability to the ISP1 router.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 capability orf prefix-
list send
```

Step 5

On the R2 router, configure the prefix list with the name FILTER, which will only permit 10.0.1.16/28 network. This prefix list will be used for prefix-based outbound route filtering.

```
R2(config)# ip prefix-list FILTER seq 5 permit 10.0.1.16/28
```

Step 6

On the R2 router, apply the "FILTER" prefix list to incoming advertisements from the ISP1 neighbor.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 prefix-list FILTER in
```

Step 7

Clear all BGP sessions on the router ISP1.

```
ISP1# clear ip bgp *
```

Step 8

On the ISP1 router, verify that the ORF capability exchange is enabled.

```
ISP1# show ip bgp neighbors 172.16.12.2
BGP neighbor is 172.16.12.2, remote AS 1, external link
  BGP version 4, remote router ID 10.0.0.65
  BGP state = Established, up for 00:15:44
  Last read 00:00:27, last write 00:00:09, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
    Multisession Capability:
    Stateful switchover support enabled: NO for session 1
  Message statistics:
    InQ depth is 0
    OutQ depth is 0

              Sent          Rcvd
Opens:             1           1
Notifications:     0           0
Updates:           2           2
Keepalives:        18          19
Route Refresh:      0           0
Total:             21          25
Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
Session: 172.16.12.2
```

```

BGP table version 15, neighbor version 15/0
Output queue size : 0
Index 28, Advertise bit 0
28 update-group member
AF-dependant capabilities:
  Outbound Route Filter (ORF) type (128) Prefix-list:
    Send-mode: received
    Receive-mode: advertised
<...output omitted...>

```

You should see that the ORF send mode capability was received and the ORF receive mode capability was advertised.

Step 9

On the ISP1 router, verify that it has received an IP prefix list that defines an outbound route filter for the 10.0.1.16/28 subnet.

```

ISP1# show ip bgp neighbors 172.16.12.2 received prefix-filter
Address family: IPv4 Unicast
ip prefix-list 172.16.12.2: 1 entries
  seq 5 permit 10.0.1.16/28

```

Step 10

On the ISP1 router, verify that BGP is advertising 10.0.1.16/28 subnet only to the R2 router.

```

ISP1# show ip bgp neighbor 172.16.12.2 advertised-routes
BGP table version is 16, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.1.16/28     0.0.0.0              0         32768 i

```

Step 11

On the R2 router, verify that only 10.0.1.16/28 was received from the ISP1 router.

```

R2# show ip bgp
BGP table version is 137, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.0.0/24       0.0.0.0              0         32768 i
*> 10.0.0.16/28      0.0.0.0              0         32768 i
*> 10.0.0.32/28      0.0.0.0              0         32768 i
*> 10.0.0.48/28      0.0.0.0              0         32768 i
*> 10.0.0.64/28      0.0.0.0              0         32768 i
*> 10.0.1.16/28      172.16.12.11         0           0 100 i
*> 10.0.2.0/28       172.16.22.22         0           0 200 i
*> 10.0.2.16/28      172.16.22.22         0           0 200 i
*> 10.0.2.32/28      172.16.22.22         0           0 200 i
*> 10.0.2.48/28      172.16.22.22         0           0 200 i
*> 10.0.2.64/28      172.16.22.22         0           0 200 i
*> 10.0.2.80/28      172.16.22.22         0           0 200 i
*> 10.0.2.96/28      172.16.22.22         0           0 200 i
*> 10.0.2.112/28     172.16.22.22         0           0 200 i
   Network          Next Hop           Metric LocPrf Weight Path
*> 10.0.2.128/28     172.16.22.22         0           0 200 i
*> 10.0.21.0/24      172.16.22.22         0           0 200 21 i
*> 10.0.37.0/24      172.16.22.22         0           0 200 21 40 37 i
*> 10.0.40.0/24      172.16.22.22         0           0 200 21 40 i

```

Summary

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

- The BGP prefix-based outbound route filtering feature uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers.
- Configuring ORF helps you to reduce the amount of system resources that are required for generating and processing routing updates by filtering out unwanted routing updates at the source.
- The ORF message contains the information that is used to determine which updates will be passed.
- The **neighbor *ip-address* capability orf prefix-list** command with the **send** and **receive** keywords enables ORF negotiations and activates an ORF prefix-list.
- Use the **clear ip bgp *neighbor*** command to trigger a BGP route refresh.
- With the **show ip bgp neighbors** command, neighbor-supported ORF capabilities are displayed as "advertised, received," and a filter of the supported type can be sent to the neighbor.
- You can verify that prefix list was received on ORF receiver using the **show ip bgp neighbors *ip-address* received prefix** command.

Applying Route Maps as BGP Filters

Overview

BGP is a powerful routing protocol that supports a wide variety of administrative policy controls and route selection features. You cannot achieve many complex filtering goals and administrative policies by using only single-purpose filtering methods or by compounding multiple filtering methods together. Route maps provide a method to perform various compound, complex filtering operations within a single tool. Understanding the operation and use of route maps is a critical component in the successful implementation of any large-scale BGP deployment.

In this lesson, you will learn about route maps and how you can use them for BGP filtering. You will learn the commands that are required to use route maps with prefix lists. You will also learn how to use prefix lists as BGP filters and how to monitor previously configured route maps.

Upon completing this lesson, you will be able to:

- Describe the high-level function of a route-map
- Describe the function of the BGP Route-Map Policy List Support feature
- Describe the function of the BGP Route-Map Continue feature
- Identify the Cisco IOS commands that are required to configure a route map to match against a prefix list
- Identify where you can apply route maps as route filters in a BGP network
- Identify the Cisco IOS command that is required to enable a route map as a BGP route filter
- Identify the Cisco IOS commands that are required to monitor the operation of a configured route map that is used as a BGP filter

Route Map Overview

A route map is a filter, and what the route map denies is dropped. Additionally, you can use the route map to modify attributes of the permitted routes.

Route maps are very complex access lists:

- Access lists have lines.
 - Route maps contain statements.
- Access lists use addresses and masks.
 - Route maps use match conditions.
- With access lists, there is an access list number.
 - With route maps, there is a route map name.
- Statements in route maps are numbered.
 - You can insert and delete statements in a route map.
 - You can edit match conditions in a statement.
- Route map statements can modify matched routes with "set" options.

Route maps are similar to access lists. Both have a set of tests to be performed, and several tests can be done in sequence. The first match produces the result of either "permit" or "deny."

An access list has a number of lines, each indicating a testing condition. The route map is more complex than the access list. The route map consists of several groups of configuration lines; each group is called a statement. The statement has a sequence number that provides the opportunity to remove or modify an explicit statement without removing the entire route map. There is also an opportunity to add a new statement between two existing statements.

Each route map statement starts with a configuration line indicating the name of the route map, the sequence number, and whether the result should be permitted or denied if the testing matches. The statement then continues, following configuration lines with the match clauses. Matching can be done in several ways: testing on the prefix, the AS path, or some other attribute. The statement concludes with optional "set" statements, where attributes may be modified or set.

```
route-map name [permit|deny sequence]
match condition
match condition
set parameter
```

- The default statement action is "permit."
- A route that is not matched by any statement is dropped.
- "Permit all" is achieved by specifying "permit" without a "match" clause.
- Match conditions in one statement are ANDed together.
- The first matching statement permits or denies the route.

A route map consists of several statements. Each statement starts with the route map configuration line, on which the name of the route map must be indicated. A good practice is to always indicate the **permit** or **deny** keyword followed by a sequence number.

The matching clauses for the statements are listed on the match lines following the route map line. There may be several match lines, each referring to a different test to be performed. All tests must be passed for the statement to be matched. If any of the match line tests fails, the next route map statement is tried. Statements are tried in sequence number order. If there are no more statements in the route map, the result is, implicitly, "deny."

If all of the match clauses succeed, there is a match for the statement and the indicated result is used. If the result is to deny, the route is then silently ignored. If the result is to permit, the route is accepted and the set clauses are applied. The set clauses allow one or more attributes to be changed or set to specific values before the route is accepted.

- Route map conditions are specified in the match statement.
- Route maps can match on:
 - Network number and subnet mask that are matched with an IP prefix list
 - Route originator

- BGP next-hop address
- BGP origin
- Tag attached to IGP route
- AS path
- BGP community that is attached to BGP route
- IGP route type (internal/external ...)

Each route map statement can have several match clauses, and each match clause is given its own configuration line. The match clause refers to the tests that will be made on the candidate route. Tests of the candidate route can be based on the following criteria:

- IP network numbers and subnet masks, by referring to a prefix list or access list that will be applied on the route.
- Route originator, by referring to a prefix list or access list that will be applied on the value of the originator [BGP](#) attribute.
- Next hop, by referring to a prefix list or access list that will be applied on the value of the next-hop BGP attribute.
- Origin code, by testing the value of the origin BGP attribute.
- Tag value that is attached to an [IGP](#) route—used only when redistribution from an IGP into BGP occurs.
- [AS](#) path, by referring to an AS path access list that will be applied on the value of the AS path BGP attribute.
- Community, by referring to a community list that will be applied on the value of the Community BGP attribute.
- IGP route type, by testing if the IGP route is internal or external—used only when redistribution from an IGP to BGP occurs.
- Route maps can also change the attributes of BGP routes.
- Route maps can set:
 - Origin
 - BGP next-hop
 - Weight
 - BGP community
 - Local preference
 - MED

Each route map statement may have several set clauses. Each set clause is applied to the route when the route map statement permits the route. With a route map, the following can be set:

- Origin BGP attribute
- Next-hop BGP attribute
- Weight
- Community BGP attribute
- Local preference BGP attribute
- [MED](#) BGP attribute, by setting the metric

BGP Route-Map Policy List Support

The BGP Route-Map Policy List Support feature introduces new functionality to [BGP](#) route maps, allowing you to group route map match clauses into named lists—policy lists. A policy list functions like a macro.

- Adds the capability for a network operator to group route map match clauses into named lists that are called policy lists.
- Simplifies the configuration of BGP routing policy in medium-size and large networks. You can preconfigure policy lists with groups of match clauses and then reference these policy lists within different route maps.
- Eliminates need to manually reconfigure each recurring group of match clauses that occur in multiple route map entries.

When a policy list is referenced in a route map, all of the match clauses are evaluated and processed as if they had been configured directly in the route map. The BGP Route Map Policy List Support feature simplifies the configuration of BGP routing policy in medium-size and large networks. A network operator can preconfigure policy lists with groups of match clauses and then reference these policy lists within different route maps. You no longer need to manually reconfigure each recurring group of match clauses in multiple route map entries.

A policy list is like a route map that contains only match clauses. The policy list is created and then referenced within a route map. There are no changes to match clause semantics and route map functions. Match clauses are configured in policy lists with permit and deny statements. The route map evaluates and processes each match clause and permits or denies routes based on the configuration. AND and OR semantics in the route map function the same way for policy lists as they do for match and set clauses. There are some commands that are related to the BGP Route-Map Policy List Support feature: the **ip policy-list** command, the **match policy-list** command, and the **show ip policy-list** command.

```
router# ip policy-list policy-list-name {permit | deny}
```

- Creates a BGP policy list

```
router# match policy-list policy-list-name
```

- Configures a route map to evaluate and process a BGP policy list in a route map

```
router# show ip policy-list policy-list-name
```

- Displays one or all filter lists

To create a BGP policy list, use the **ip policy-list** command in the policy map configuration mode.

```
ip policy-list policy-list-name {permit | deny}
```

To remove a policy-list, use the **no** form of this command

```
no ip policy-list policy-list-name
```

Syntax Description

Parameter	Description
<i>policy-list-name</i>	Name of the configured policy list
permit	Permits access for matching conditions
deny	Denies access to matching conditions

To configure a route map to evaluate and process a BGP policy list in a route map, use the **match policy-list** command in the route map configuration mode.

```
match policy-list policy-list-name
```

To remove a path list entry, use the **no** form of this command.

```
no match policy-list policy-list-name
```

Syntax Description

Parameter	Description
<i>policy-list-name</i>	Name of the configured policy list

To display information about a configured policy list and policy list entries, use the **show ip policy-list** command in the user EXEC mode.

```
show ip policy-list policy-list-name
```

Syntax Description

Parameter	Description
<i>policy-list-name</i>	Name of the configured policy list

Configuring Policy List Examples

The following configuration example creates a BGP policy list that permits matches on the [AS](#) path and [MED](#) of a router:

```
Router(config)# ip policy-list POLICY-LIST-NAME-1 permit
Router(config-policy-list)# match as-path 1
Router(config-policy-list)# match metric 10
Router(config-policy-list)# end
```

The following configuration example creates a BGP policy list that permits matches on the specified BGP community and the next hop of a router:

```
Router(config)# ip policy-list POLICY-LIST-NAME-2 permit
Router(config-policy-list)# match community 20
Router(config-policy-list)# match metric 10
Router(config-policy-list)# ip community-list 20 permit 20:1
Router(config-policy-list)# end
```

The following configuration example creates a BGP policy list that denies matches on the specified BGP community and the next hop of a router:

```
Router(config)# ip policy-list POLICY-LIST-NAME-3 deny
Router(config-policy-list)# match community 20
Router(config-policy-list)# match metric 10
Router(config-policy-list)# end
```

Configuring Route Maps to Reference Policy List Examples

The configuration examples in this section create BGP route maps that reference BGP policy lists with the **route-map** route map configuration command.

The following configuration example creates a route map that references policy lists and separate match and set clauses in the same configuration. This example uses AND semantics between POLICY-LIST-NAME-1 and POLICY-LIST-NAME-2.

```
Router(config)# route-map MAP-NAME-1 10
Router(config-route-map)# match ip-address 1
Router(config-route-map)# match policy-list POLICY-LIST-NAME-1
Router(config-route-map)# match policy-list POLICY-LIST-NAME-2
Router(config-route-map)# set community 10:1
Router(config-route-map)# set local-preference 140
Router(config-route-map)# end
```

The following configuration example creates a route map that references policy lists and separate match and set clauses in the same configuration. This example uses OR semantics between POLICY-LIST-NAME-3 and POLICY-LIST-NAME-4.

```
Router(config)# route-map MAP-NAME-2 10
Router(config-route-map)# match policy-list POLICY-LIST-NAME-3 POLICY-
LIST-NAME-4
Router(config-route-map)# set community 10:1
Router(config-route-map)# set local-preference 140
```

```
Router(config-route-map)# end
```

Verifying BGP Route-Map Policy List Support

To verify that a policy list has been created, use the **show ip policy-list** command. The output of this command displays the policy list name and configured match clauses. The following sample output is similar to the output that will be displayed:

```
Router# show ip policy-list
policy-list POLICY-LIST-NAME-1 permit
  Match clauses:
    metric 20
policy-list POLICY-LIST-NAME-2 permit
  Match clauses:
    as-path (as-path filter): 1
```



A policy list name can be specified when the **show ip policy-list** command is entered. This option can be useful for filtering the output of this command and verifying a single policy list

To verify that a route map has been created and a policy list is referenced, use the **show route-map** command. The output of this command displays the route map name and policy-lists that the configured route maps reference. The following sample output is similar to the output that will be displayed:

```
Router# show route-map
route-map ROUTE-MAP-NAME-1, deny, sequence 10
  Match clauses:
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME-1, permit, sequence 10
  Match clauses:
    IP Policy lists:
      POLICY-LIST-NAME-1
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
```

BGP Route Map Continue

The BGP Route-Map Continue feature introduces the continue clause to the BGP route map configuration. The continue clause provides more programmable policy configuration and route filtering. It introduces the ability to execute extra entries in a route map after an entry is executed with successful match and set clauses. Continue clauses allow you to configure and organize more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map.

- Introduces the continue clause to the BGP route map configuration, providing more programmable policy configuration and route filtering
- Enables you to execute extra entries in a route map after an entry is executed with successful match and set clauses
- Allows configuration and organization of more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map
- Allows modularization of network policy configuration so that repeated policy definitions can be reduced within the same route map

Continue clauses provide a programmable method to organize and control the flow of a route map. Route map configuration was linear before this feature was introduced. Continue clauses also allow you to modularize network policy configuration so that repeated policy definitions can be reduced within the same route map.

Route Map Operation Without Continue Clauses

A route map evaluates match clauses until a successful match occurs. After the match occurs, the route map stops evaluating match clauses and starts executing set clauses, in the order in which they were configured. If a successful match does not occur, the route map "falls through" and evaluates the next sequence number of the route map. It repeats until all configured route map entries have been evaluated or a successful match occurs. Each route map sequence is tagged with a sequence number to identify the entry. Route map entries are evaluated in order, starting with the lowest sequence number and ending with the highest sequence number. If the route map contains only set clauses, the set clauses are executed automatically, and the route map does not evaluate any other route map entries.

Route Map Operation with Continue Clauses

When a continue clause is configured, the route map continues to evaluate and execute match clauses in the specified route map entry after a successful match occurs. The continue clause can be configured to go to (or jump to) a specific route map entry by specifying the sequence number. Or, if a sequence number is not specified, to go to the next sequence number. This behavior is called an "implied continue." If a match clause exists, the continue clause is executed only if a match occurs. If no successful matches occur, the continue clause is ignored.

If a match clause does not exist in the route map entry but a continue clause does, the continue clause is automatically executed and goes to the specified route map entry. If a match clause exists in a route map entry, the continue clause is executed only when a successful match occurs. When a successful match occurs and a continue clause exists, the route map executes the set clauses and then goes to the specified route map entry. If the next route map contains a continue clause, the route map executes the continue clause if a successful match occurs. If a continue clause does not exist in the next route map, the route map is evaluated normally. If a continue clause exists in the next route map but a match does not occur, the route map does not continue. It falls through to the next sequence number, if one exists.



A continue clause can be executed without a successful match if a route map entry does not contain a match clause.

```
router# continue sequence-number
```

- Configures a route map to go to a route map entry with a higher sequence number

```
router# show route-map [map-name]
```

- Displays configured route maps

You will use two commands with the BGP Route-Map Continue feature, the **continue** command and the **show route-map** command.

To configure a route map to go to a route map entry with a higher sequence number, use the **continue** command in route map configuration mode.

```
continue sequence-number
```

To remove a continue clause from a route map, use the **no** form of this command.

```
no continue
```

Syntax Description

Parameter	Description
<i>sequence-number</i>	(Optional) Route map sequence number. If a route map sequence number is not specified when configuring a continue clause, the continue clause continues to the route map entry with the next sequence number. This behavior is referred to as an "implied continue"

To display the configured route maps, use the **show route-map** command in the EXEC mode.

```
show route-map [map-name]
```

Syntax Description

Parameter	Description
<i>map-name</i>	(Optional) Name of a specific route map

BGP Route-Map Continue Clause Configuration Example

The following example shows the continue clause configuration in a route map sequence.

The first continue clause in route map entry 10 indicates that the route map will go to route map entry 30 if a successful match occurs. If a match does not occur, the route map will fall through to route map entry 20. If a successful match occurs in route map entry 20, the set action will be executed and the route map will not evaluate any additional route map entries.

If a successful match does not occur in route map entry 20, the route map will fall through to route map entry 30. This sequence does not contain a match clause, so the set clause will be automatically executed. The continue clause will go to the next route map entry because a sequence number is not specified.

If there are no successful matches, the route map will fall through to route map entry 30 and execute the set clause, and route map entry 40 will not be evaluated.

```
route-map ROUTE-MAP-NAME permit 10
match ip address 1
match metric 10
set as-path prepend 10
continue 30
!
route-map ROUTE-MAP-NAME permit 20
match ip address 2
match metric 20
set as-path prepend 10 10
!
route-map ROUTE-MAP-NAME permit 30
set as-path prepend 10 10 10
continue
!
route-map ROUTE-MAP-NAME permit 40
match community 10:1
set local-preference 104
```

BGP Route-Map Continue Clause Verification Example

To verify the configuration of continue clauses, use the **show route-map** command. The output of this command displays configured route maps, match, set, and continue clauses. The following sample output is similar to the output that will be displayed:

```
Router# show route-map
route-map ROUTE-MAP-NAME, permit, sequence 10
Match clauses:
  ip address (access-lists): 1
  metric 10
Continue: sequence 40
Set clauses:
  as-path prepend 10
Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME, permit, sequence 20
Match clauses:
  ip address (access-lists): 2
  metric 20
Set clauses:
  as-path prepend 10 10
Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME, permit, sequence 30
Match clauses:
Continue: to next entry 40
Set clauses:
  as-path prepend 10 10 10
Policy routing matches: 0 packets, 0 bytes
route-map ROUTE-MAP-NAME, permit, sequence 40
Match clauses:
  community (community-list filter): 10:1
Set clauses:
  local-preference 104
Policy routing matches: 0 packets, 0 bytes
route-map LOCAL-POLICY-MAP, permit, sequence 10
Match clauses:
Set clauses:
  community 655370
Policy routing matches: 0 packets, 0 bytes
```

Prefix List Use in Route Maps

You can use prefix lists as match criteria in several matching statements in the route map.

```
router(config-route-map)# match ip address prefix-list list-name
```

- Uses prefix list to match routes in route-map match condition

```
router(config-route-map)#match ip next-hop prefix-list list-name
```

- Matches routes where the next hop matches the conditions in the prefix list

```
router(config-route-map)# match ip route-source prefix-list list-name
```

- Matches routes that are received from BGP peer that matches the prefix list

You can use the **match ip address** command in the route map configuration mode to perform policy routing on packets. You can use the same command to distribute any routes that have a destination network number address that a standard access list, an extended access list, or a prefix list permits.

```
match ip address {access-list-number [access-list-number... | access-list-name...] | access-list-name [access-list-number... | access-list-name] | prefix-list prefix-list-name [prefix-list-name...]}
```

To remove the **match ip address** entry, use the **no** form of this command.

```
no match ip address {access-list-number [access-list-number... | access-list-name...] | access-list-name [access-list-number... | access-list-name] | prefix-list prefix-list-name [prefix-list-name...]}
```

Syntax Description

Parameter	Description
<i>access-list-number...</i>	Number of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.
<i>access-list-name...</i>	Name of a standard or extended access list. It can be an integer from 1 to 199. The ellipsis indicates that multiple values can be entered.
prefix-list	Distributes routes based on a prefix list.
<i>prefix-list-name...</i>	Name of a specific prefix list. The ellipsis indicates that multiple values can be entered.

To redistribute any routes that have a next-hop router address that is passed by one of the specified access lists, use the **match ip next-hop** command in the route map configuration mode.

```
match ip next-hop {access-list-number | access-list-name}[...access-list-number | ...access-list-name]
```

To remove the next hop entry, use the **no** form of this command.

```
no match ip next-hop {access-list-number | access-list-name}[...access-list-number | ...access-list-name]
```

Syntax Description

Parameter	Description
<i>access-list-number</i>	Number of a standard or extended access-list. It can be an integer from 1 to 199.
<i>access-list-name</i>	Name of a standard or extended access-list. It can be an integer from 1 to 199.

To redistribute routes that have been advertised by routers and access servers at

the address that the access lists specify, use the **match ip route-source** command in the route map configuration mode.

```
match ip route-source {access-list-number | access-list-name}[...access-list-number | ...access-list-name]
```

To remove the route-source entry, use the **no** form of this command.

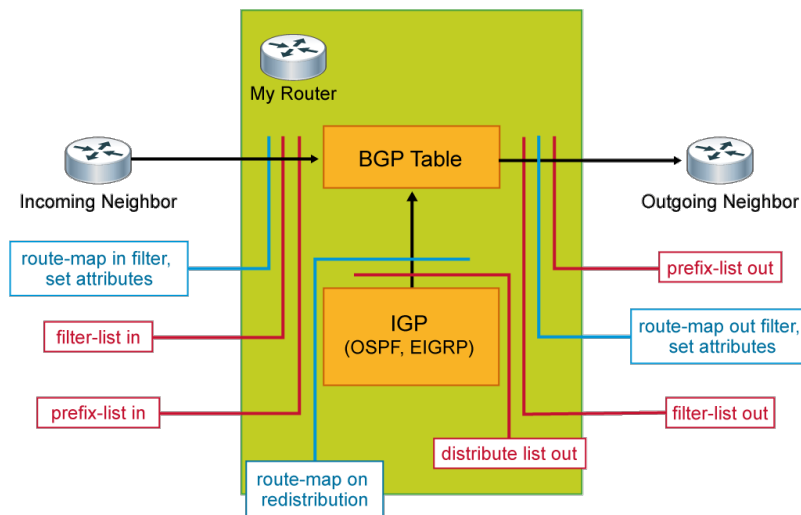
```
no match ip route-source {access-list-number | access-list-name}[...access-list-number | ...access-list-name]
```

Syntax Description

Parameter	Description
<i>access-list-number</i>	Number of a standard or extended access list. It can be an integer from 1 to 199.
<i>access-list-name</i>	Name of a standard or extended access list. It can be an integer from 1 to 199.

BGP Filters

You can optionally apply filter lists, prefix lists, and route maps to either incoming or outgoing information or any combination of the two.



The incoming prefix list, the incoming filter list, and the incoming route map must all permit the routes that are received from a neighbor before being accepted into the BGP table. Outgoing routes must pass the outgoing filter list, the outgoing prefix list, and the outgoing route map before being transmitted to the neighbor. When a router is configured to redistribute routing information from an IGP into BGP, the routes must successfully pass any prefix list or route map that is applied to the redistribution. This happens before a route is injected into the BGP table.

Using Route Maps as BGP Filters

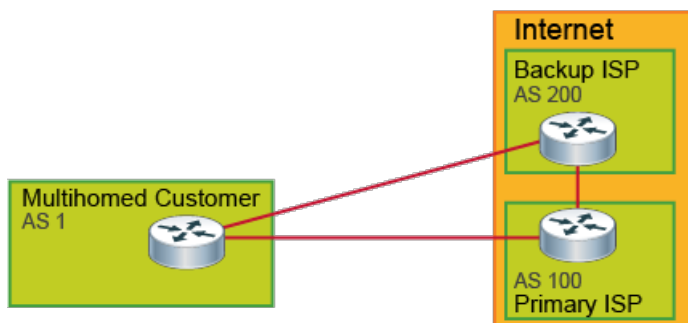
You can apply a route map on incoming or outgoing routing information for a neighbor. The route map must permit the routing information in order to be accepted. If there is no statement in the route map explicitly permitting a route, then the route will be implicitly denied and dropped.

```
router(config-router)# neighbor ip-address route-map name [in | out]
```

- This command applies a route map to incoming or outgoing BGP updates.
- Prefixes not permitted by the route map are discarded.
- Route maps can also change BGP attributes in incoming or outgoing updates.
- Route maps, filter lists, and prefix lists are evaluated in sequence (effectively ANDed together).

The permitted routes may have their attributes set or changed by the set clauses in the route map. Setting attributes on routes is useful when you are influencing the route selection. One of the statements in the route map can permit some routes and change their attributes. Another statement in the route map could permit other routes and not alter their attributes. When route selection is performed, the attribute values indicate that one route is preferred over the other.

- **Requirement:** The customer will accept only a default route from both service providers.



```
router bgp 1
 neighbor Primary-ISP remote-as 100
 neighbor Primary-ISP route-map filter in
 neighbor Secondary-ISP remote-as 100
 neighbor Secondary-ISP route-map filter in
!
route-map filter permit 10
 match ip address prefix-list DefaultOnly
!
ip prefix-list DefaultOnly seq 10 permit 0.0.0.0/0
```

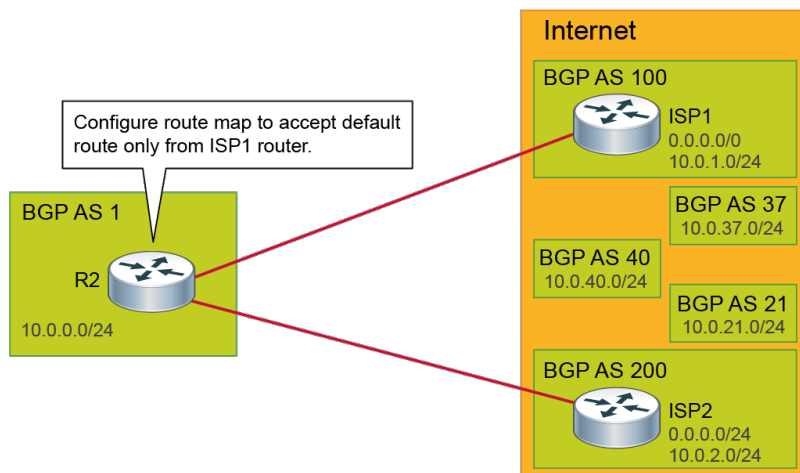
In this example, the customer will accept only a default route from both service providers. The customer does not want to put any other network into the [BGP](#) table than the default route. The default route is announced via BGP to the customer from both service providers.

In the example, the route map with a name *filter* is applied in inbound direction on the customer router for both service provider neighbors. The route map is matching the prefix list, which permits default route only.

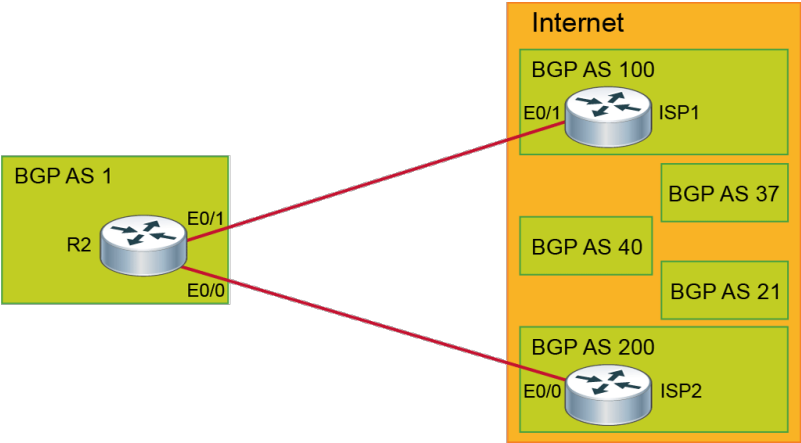
Discovery 10: Configure Route Maps as BGP Filters

Overview

Through this discovery, you will learn how to configure route maps to filter [BGP](#) updates. You as a customer configuring router R2 will accept only the default route that is announced via BGP from the ISP1 router. You will not accept any other networks that are announced via BGP from the service provider.



Topology



Job Aids

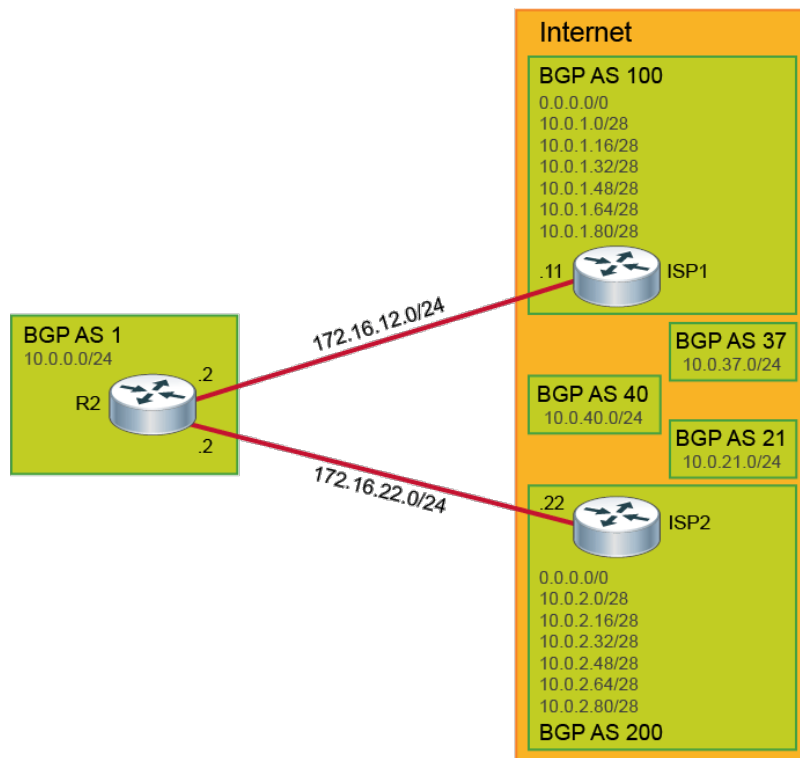


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP1	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate extra networks in different autonomous systems.
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks
ISP2	Loopback 21 Loopback 37 Loopback 40	10.0.21.1 10.0.37.1 10.0.40.1	Loopbacks simulate extra networks in different autonomous systems.

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#) (R2 to ISP1 and R2 to ISP2). The ISP1 router announces these networks:

- Default route 0.0.0.0/0
- 10.0.1.16/28
- 10.0.1.32/28
- 10.0.1.48/28
- 10.0.1.64/28
- 10.0.1.80/28

Configure Route Maps as BGP Filters

Discovery Steps

Step 1

On the ISP1 router, verify that the default route is being announced in an EBGP session to the customer router R2.

```
ISP1# show ip bgp neighbor 172.16.12.2 advertised-routes
BGP table version is 13, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Originating default network 0.0.0.0

   Network        Next Hop           Metric LocPrf Weight Path
*> 10.0.1.0/28     0.0.0.0             0         32768 i
*> 10.0.1.16/28    0.0.0.0             0         32768 i
*> 10.0.1.32/28    0.0.0.0             0         32768 i
*> 10.0.1.48/28    0.0.0.0             0         32768 i
*> 10.0.1.64/28    0.0.0.0             0         32768 i
*> 10.0.1.80/28    0.0.0.0             0         32768 i
*> 10.0.21.0/24    0.0.0.0             0         32768 i
*> 10.0.37.0/24    0.0.0.0             0         32768 i
*> 10.0.40.0/24    0.0.0.0             0         32768 i
```

You can see that the ISP1 router originates the default route.

Step 2

On the ISP2 router, verify that the default route is being announced in an EBGP session to the customer router R2.

```
ISP2# show ip bgp neighbors 172.16.22.2 advertised-routes
BGP table version is 20, local router ID is 10.0.40.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

Originating default network 0.0.0.0

   Network        Next Hop           Metric LocPrf Weight Path
*> 10.0.2.0/28     0.0.0.0             0         32768 i
*> 10.0.2.16/28    0.0.0.0             0         32768 i
*> 10.0.2.32/28    0.0.0.0             0         32768 i
*> 10.0.2.48/28    0.0.0.0             0         32768 i
*> 10.0.2.64/28    0.0.0.0             0         32768 i
*> 10.0.2.80/28    0.0.0.0             0         32768 i
*> 10.0.2.96/28    0.0.0.0             0         32768 i
*> 10.0.2.112/28   0.0.0.0             0         32768 i
*> 10.0.2.128/28   0.0.0.0             0         32768 i
*> 10.0.21.0/24    0.0.0.0             0         32768 i
*> 10.0.37.0/24    0.0.0.0             0         32768 i
*> 10.0.40.0/24    0.0.0.0             0         32768 i
   Network        Next Hop           Metric LocPrf Weight Path

Total number of prefixes 12
```

You can see that the ISP2 router originates the default route.

Step 3

On the R2 router, verify that the customer router R2 receives default route from both neighbors, the ISP1 and ISP2 routers. You should also make sure that the R2 router receives several other routes from both [ISP](#) routers.

```
R2# show ip bgp
BGP table version is 26, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop           Metric LocPrf Weight Path
* 0.0.0.0         172.16.22.22        0         200 i
*> 172.16.12.11    172.16.12.11        0         100 i
*> 10.0.0.0/24     0.0.0.0             0         32768 i
*> 10.0.0.16/28    0.0.0.0             0         32768 i
*> 10.0.0.32/28    0.0.0.0             0         32768 i
*> 10.0.0.48/28    0.0.0.0             0         32768 i
*> 10.0.0.64/28    0.0.0.0             0         32768 i
*> 10.0.1.0/28     172.16.12.11        0         100 i
*> 10.0.1.16/28    172.16.12.11        0         100 i
*> 10.0.1.32/28    172.16.12.11        0         100 i
*> 10.0.1.48/28    172.16.12.11        0         100 i
*> 10.0.1.64/28    172.16.12.11        0         100 i
*> 10.0.1.80/28    172.16.12.11        0         100 i
*> 10.0.2.0/28     172.16.22.22        0         200 i
   Network        Next Hop           Metric LocPrf Weight Path
*> 10.0.2.16/28    172.16.22.22        0         200 i
*> 10.0.2.32/28    172.16.22.22        0         200 i
*> 10.0.2.48/28    172.16.22.22        0         200 i
*> 10.0.2.64/28    172.16.22.22        0         200 i
*> 10.0.2.80/28    172.16.22.22        0         200 i
*> 10.0.2.96/28    172.16.22.22        0         200 i
*> 10.0.2.112/28   172.16.22.22        0         200 i
```


10.0.2.128/28	172.16.22.22	0	0 200 1
*> 10.0.2.128/28	172.16.22.22	0	0 200 1
*> 10.0.21.0/24	172.16.22.22	0	0 200 21 i
* 10.0.21.0/24	172.16.22.22	0	0 100 37 40 21 i
* 10.0.37.0/24	172.16.22.22	0	0 200 21 40 37 i
*> 10.0.37.0/24	172.16.22.22	0	0 100 37 i
* 10.0.40.0/24	172.16.22.22	0	0 200 21 40 i
*> 10.0.40.0/24	172.16.22.22	0	0 100 37 40 i

You can see that the default route on the R2 router was received from two different EBGP neighbors:

- One default route was received from the ISP1 router with the AS path attribute set to 100 and BGP next hop set to 172.16.12.11. This address is the IP address of the ISP1 router.
- Another default route was received from the ISP2 router with the AS path attribute set to 200 and BGP next hop set to 172.16.22.22. This address is the IP address of ISP2 router.

You should receive these BGP routes on the R2 router:

- Any subnets in 10.0.1.0/24 range from the ISP1 router
- Any subnets in 10.0.2.0/24 range from the ISP2 router
- Networks 10.0.21.0/24, 10.0.37.0/24, 10.0.40.0/24 from both the ISP1 and ISP2 routers

Step 4

Configure the route map on the R2 router. The route map should permit only the default route, that is defined in the prefix list, to be accepted from the ISP1 EBGp neighbor.

```
R2(config)# ip prefix-list DefaultOnly permit 0.0.0.0/0
R2(config)# route-map ISP1 permit 10
R2(config-route-map)# match ip address prefix-list DefaultOnly
R2(config-route-map)# exit
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 route-map ISP1 in
```

Step 5

Enable debugging of BGP updates on the R2 router.

```
R2# debug ip bgp update
```

Step 6

Clear all BGP sessions on the R2 router.

```
R2# clear ip bgp *
```

Step 7

On the R2 router, observe that all non default route BGP updates received from the ISP1 router were denied because of the configured route map. Only the default route that is received via the EBGP session from the ISP1 router should be installed to the routing table.

```
BGP-5-ADJCHANGE: neighbor 172.16.12.11 Down User reset
*Mar 6 00:36:02.681: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.12.11 IPv4 Unicast topology base removed from session User reset
*Mar 6 00:36:02.681: %BGP-5-ADJCHANGE: neighbor 172.16.22.22 Down User reset
*Mar 6 00:36:02.681: %BGP_SESSION-5-ADJCHANGE: neighbor 172.16.22.22 IPv4 Unicast topology base removed from session User reset
*Mar 6 00:36:03.107: %BGP-5-ADJCHANGE: neighbor 172.16.12.11 Up
*Mar 6 00:36:03.109: %BGP-5-ADJCHANGE: neighbor 172.16.22.22 Up
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd UPDATE w/ attr: nexthop 172.16.12.11, origin i, metric 0, merged path 100 37 40, AS_PATH
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.40.0/24 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd UPDATE w/ attr: nexthop 172.16.12.11, origin i, metric 0, merged path 100 37, AS_PATH
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.37.0/24 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd UPDATE w/ attr: nexthop 172.16.12.11, origin i, metric 0, merged path 100 37 40 21, AS_PATH
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.21.0/24 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd UPDATE w/ attr: nexthop 172.16.12.11, origin i, metric 0, merged path 100, AS_PATH
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.0/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.16/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.32/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.48/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.64/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.113: BGP(0): 172.16.12.11 rcvd 10.0.1.80/28 -- DENIED due to: route-map;
*Mar 6 00:36:03.114: BGP(0): 172.16.12.11 rcvd UPDATE w/ attr: nexthop 172.16.12.11, origin i, merged path 100, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.12.11 rcvd 0.0.0.0/0
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i, metric 0, merged path 200 21 40, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.40.0/24
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i, metric 0, merged path 200 21 40 37, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.37.0/24
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i, metric 0, merged path 200 21, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.21.0/24
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i, metric 0, merged path 200, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.0/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.16/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.32/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.48/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.64/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.80/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.96/28
R2#
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.112/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 10.0.2.128/28
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 172.16.22.22, origin i, merged path 200, AS_PATH
*Mar 6 00:36:03.114: BGP(0): 172.16.22.22 rcvd 0.0.0.0/0
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 0.0.0.0/0 -> 172.16.12.11(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.0/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.16/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.32/28 -> 172.16.22.22(global) to main IP table
```

```
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.48/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.64/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.80/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.96/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.112/28 -> 172.16.22.22(global) to main IP table
*Mar 6 00:36:04.130: BGP(0): Revise route installing 1 of 1 routes for 10.0.2.128/28 -> 172.16.22.22(global) to main IP table
```

Step 8

On the R2 router, verify the content of BGP table.

```
R2# show ip bgp
BGP table version is 19, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*    0.0.0.0          172.16.22.22              0 200 i
*>   172.16.12.11          172.16.12.11              0 100 i
*>   10.0.0.0/24         0.0.0.0                  0      32768 i
*>   10.0.0.16/28        0.0.0.0                  0      32768 i
*>   10.0.0.32/28        0.0.0.0                  0      32768 i
*>   10.0.0.48/28        0.0.0.0                  0      32768 i
*>   10.0.0.64/28        0.0.0.0                  0      32768 i
*>   10.0.2.0/28         172.16.22.22              0      0 200 i
*>   10.0.2.16/28        172.16.22.22              0      0 200 i
*>   10.0.2.32/28        172.16.22.22              0      0 200 i
*>   10.0.2.48/28        172.16.22.22              0      0 200 i
*>   10.0.2.64/28        172.16.22.22              0      0 200 i
*>   10.0.2.80/28        172.16.22.22              0      0 200 i
*>   10.0.2.96/28        172.16.22.22              0      0 200 i
   Network          Next Hop        Metric LocPrf Weight Path
*>   10.0.2.112/28       172.16.22.22              0      0 200 i
*>   10.0.2.128/28       172.16.22.22              0      0 200 i
*>   10.0.21.0/24        172.16.22.22              0      0 200 21 i
*>   10.0.37.0/24        172.16.22.22              0      0 200 21 40 37 i
*>   10.0.40.0/24        172.16.22.22              0      0 200 21 40 i
```

You should only see the default route that is installed for BGP updates from the ISP1 router with next hop IP address 172.16.12.11. All other networks were received from the ISP2 router.

Displaying Routes Matching the Route Map

Displaying Routes Matching the Route Map

```
router# show ip bgp route-map route-map-name
```

- Displays all routes in BGP table matching the route map
- Used for filtering the **show ip bgp** output on basis of BGP path attributes:
 - Community
 - Local preference
 - Weight
 - Origin
 - Next-hop
- Can also filter based on prefixes
- Allows powerful combined filtering

You can also use route maps for selective and powerful filtering of the BGP table. The **show ip bgp route-map** command displays the selected routes from a BGP routing table based on the contents of a route map.

A route map can match the routes based on BGP path attributes (local preference, community, weight, origin, next-hop) or prefix lists and access lists (matching IP prefixes). The power of route map filtering lies in the possibility of combining different filters (for example, filtering on community, prefix, and next-hop values).

Step 9

Configure the route map on the R2 router. The route map should permit only networks passing through the autonomous system 40.

```
R2(config)# ip as-path access-list 2 permit _40_
R2(config)# route-map PassAS40 permit 10
R2(config-route-map)# match as-path 2
```

Step 10

On the R2 router, display the content of BGP table matching the configured route map.

```
R2# show ip bgp route-map PassAS40
BGP table version is 19, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*>   10.0.37.0/24        172.16.22.22              0      0 200 21 40 37 i
*>   10.0.40.0/24        172.16.22.22              0      0 200 21 40 i
```

You should only see the networks containing "40" in the AS path.

Summary

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

- A route map is a filter that has the ability to drop denied routes as well as modify attributes of the permitted routes.
- The BGP Route-Map Policy List Support feature introduces new functionality to BGP route maps. It adds the ability for a network operator to group route map match clauses into named lists that are called policy lists.
- The BGP Route-Map Continue feature introduces the continue clause to BGP route map configuration. Continue clauses provide a programmable method to organize and control the flow of a route map.
- You can configure a route map to match against a prefix list by using the **match ip address**, **match ip next-hop**, and **match ip route-source** commands.
- Filter lists, prefix lists, and route maps can optionally all be applied on either incoming or outgoing information in any combination.
- A route map can be applied on incoming or outgoing routing information to or from a neighbor. However, the route map must permit the routing information in order to be accepted.
- Monitoring route maps is possible using the **show ip bgp** and **debug ip bgp update** commands.

Implementing Changes in BGP Policy

Overview

Because of the huge volumes of routing information that [BGP](#) is capable of handling, traditional routing update methods are not feasible. Routing policies for a BGP neighbor may include filtering mechanisms such as route maps, distribute lists, prefix lists, and [AS](#) path filter lists. Each of these filters may affect inbound or outbound routing table updates.

Whenever there is an administrative change in a routing policy, the BGP session must be reset before the new policy can take effect. To accomplish this task, you can use two types of reset: hard reset and soft reset. Clearing a BGP session using a hard reset invalidates the cache and results in a negative impact on the operation of networks, because the information in the cache becomes unavailable. A soft reset is recommended because it allows routing tables to be reconfigured and activated without clearing the BGP session.

This lesson discusses routing updates in a BGP environment and the traditional methods of forcing BGP route updates after changes in a filter policy. The function and benefits of soft reconfiguration and route refresh are also discussed. The lesson also presents the commands that are required to perform a soft reconfiguration and route refresh and explains how to monitor and troubleshoot these features.

Upon completing this lesson, you will be able to:

- Identify the limitations of the traditional methods of forcing BGP route updates after changing a filter policy
- Describe the function of the BGP Soft Reset Enhancement feature
- Describe the function and benefits of the route refresh function
- Identify the Cisco IOS command that is required to trigger a route refresh
- Identify the Cisco IOS commands that are required to monitor route refresh operation

Traditional Filtering Limitations

[BGP](#) can potentially handle huge volumes of routing information. But it cannot handle it when you change configuration lines in filters or route maps. In this case, the router cannot go through the huge table of BGP information and calculate which entry is no longer valid in the local table. Nor can the router determine which route or routes, already advertised, should be withdrawn from a neighbor. There is an obvious risk that the first configuration change will be immediately followed by a second, which would cause the whole process to start all over again.

- All filters apply only to new incoming and outgoing updates.
- To change the outbound routing policy, you have to resend BGP updates to your neighbors.
- To change the inbound routing policy, you have to force your neighbor to resend the updates to you.
- The traditional mechanism is to clear BGP sessions.

To avoid such a problem, Cisco IOS software applies changes only on the updates that are received or transmitted after the configuration change has been performed. This approach means that the new routing policy, which the new filters enforce, is applied only on routes that are received or sent after the change. If network administrators would like to apply the policy change on all routes, they have to force the router to let all routes pass through the new filter.

If the filter is applied to outgoing information, the router has to resend the entire BGP table through the new filter. If the filter is applied to incoming information, the router needs its neighbor to resend its entire BGP table so that it passes through the new filters.

Traditionally, to accomplish these goals, you have torn down the affected BGP sessions after completing a configuration change. After the sessions are down, all information that is received on those sessions is invalidated and removed from the BGP table. Also, the remote neighbor will detect a session down state, and it likewise will invalidate the routes that are received on the session. After a period of 30 to 60 seconds, the sessions are re-established automatically and the entire BGP table is exchanged again, but through the new filters. This process, however, disrupts packet forwarding.

Traditional Limitations of Clearing the BGP Session

```
router# clear ip bgp {* | ip-address | peer-group-name}
```

- This command tears down the BGP session with all neighbors, a specific neighbor, or all neighbors in a peer group.
- All BGP routes are lost after the session is torn down; connectivity through the BGP neighbor is lost.
- A new session is re-established within 30 to 60 seconds.
- A full routing update is exchanged once the session is re-established, resulting in enforcement of new routing policy.
- Processing the full Internet routing table can take a long time.
- **Clearing the BGP session is a very disruptive way to implement routing policies.**

The EXEC command **clear ip bgp** tears down one or several BGP sessions. The BGP sessions are terminated, and the [TCP](#) connections closed. The neighbors go into the Idle state and stay there for approximately 30 seconds. Next, the neighbor session goes into the Active state, and the sessions are re-established.

You can implement the **clear ip bgp** command with the * (asterisk) argument, which applies to all sessions. You can also make a reference to a specific session or group of sessions to tear down.

When the session is down, all routes that are received over the session by both routers are invalidated. When the session is once again in the Established state, all BGP routes have to be resent by both peers. All routes also have to pass through the new filters, which enforces the new policy.

Exchanging the complete Internet routing table takes time, bandwidth, and [CPU](#) resources. IP packet forwarding to and from the neighbor is down for several

minutes. Also, revoking and reannouncing the routes will be registered by the rest of the Internet as a flap for each route.

BGP Soft Reset Enhancement

Previously, to perform a soft reset for inbound routing table updates, the **neighbor soft-reconfiguration** command directed the Cisco IOS software in the local BGP router to store all received (inbound) routing policy updates without modification. This method is memory-intensive and not recommended unless absolutely necessary. (Outbound updates have never required the extra memory and this feature does not affect them.)

- BGP Soft Reset Enhancement provides automatic support for dynamic soft reset of inbound BGP routing table updates that is not dependent upon stored routing table update information.
- Requires no preconfiguration (as with the **neighbor soft-reconfiguration** command).
- Requires much less memory than the previous soft reset method for inbound routing table updates.

Whenever there is a change in the routing policy, the BGP session must be cleared, or reset, for the new policy to take effect. There are two types of reset, hard reset, and soft reset. Clearing a BGP session using a hard reset invalidates the cache and results in a negative impact on the operation of networks as the information in the cache becomes unavailable. Soft reset is recommended because it allows routing tables to be reconfigured and activated without clearing the BGP session. Soft reset is done on per-neighbor basis. There are two types of soft resets:

- **Dynamic inbound soft reset:** When soft reset is used to generate inbound updates from a neighbor
- **Outbound soft reset:** When soft reset is used to send a new set of updates to a neighbor.

The BGP Soft Reset Enhancement feature, however, provides automatic support for dynamic soft reset of inbound BGP routing table updates that is not dependent on stored routing table update information. The new method requires no preconfiguration (as with the **neighbor soft-reconfiguration** command) and requires much less memory than the previous soft reset method for inbound routing table updates.

There are a number of benefits to the BGP Soft Reset Enhancement feature:

- **Allows dynamic route refresh requests:** This feature provides a way to initiate nondisruptive routing policy changes. It achieves this by allowing the dynamic exchange of route refresh requests between BGP routers, and the subsequent readvertisement of the respective outbound routing tables.
- **Requires no preconfiguration:** Because support for the soft reset using the route refresh capability is included in this release of the Cisco IOS software, no further router configuration is required. You can initiate a soft inbound reset using only the **clear ip bgp in** command.
- **Requires no additional memory resources:** Unlike a soft reset using the stored inbound routing table updates provided by the **neighbor soft-reconfiguration** command, when both BGP peers support the route refresh capability, inbound routing table updates are not stored in the local BGP router. The soft reset requests are exchanged dynamically, and no additional memory is required.
- **Provides flexibility:** There are now two available methods for inbound soft reset. The older method, using stored inbound routing table updates, and the method that this feature provides, using dynamic exchange of update information.

When the routing policy of a BGP neighbor changes, the session must be reset (cleared) for the changes to take effect. Because resetting a BGP session can be disruptive to networks, a soft reset method is recommended for reconfiguring the routing table. Previously, in order to reconfigure the inbound routing table, both the local BGP router and the BGP peer first needed to be configured to store incoming routing policy updates. BGP soft reconfiguration used to be configured using the **neighbor soft-reconfiguration** command. More resources, particularly memory, were required to store the inbound routing table updates. The **clear ip bgp** command could then initiate the soft reset, which generated a new set of inbound routing table updates using the stored information.

The BGP Soft Reset Enhancement feature provides an additional method for soft

reset. It allows the dynamic exchange of route refresh requests and routing information between BGP routers and the subsequent readvertisement of the respective outbound routing table. Soft reset using the route refresh capability does not require preconfiguration and consumes no additional memory resources.

To use this new method, both BGP peers must support the soft route refresh capability. This capability is advertised in the Open message that is sent when a peer sends its routing table update. Any router running BGP with this software release automatically supports the route refresh capability. Routers running earlier Cisco IOS software releases do not support the route refresh capability and must use the older soft reset method. If the soft reset fails, you can still clear the BGP session. However, it will have a negative impact on network operations and should be used only as a last resort.



Outbound resets have never required preconfiguration or storing of routing table updates and remain unchanged by the BGP Soft Reset Enhancement feature.

Route Refresh

Route refresh is one of the capabilities of [BGP](#). Routers use the route refresh feature to request a neighbor to resend all the routing information when it is needed.

- Route refresh is a BGP capability.
- It is used to request a neighbor to resend routing information.
- It is typically used after configuration changes to update the BGP table (route map, distribute list, prefix list, filter list, weight, local preference, MED, etc.).
- The traditional way of accomplishing this function is to clear the BGP session.
- Inbound soft reconfiguration consumes memory on the receiving router.
 - It is needed only because there is no mechanism in standard BGP to request retransmission of BGP routes.
- BGP route refresh is an optional BGP capability that allows a BGP router to request retransmission of BGP routes from a neighbor.

There are several ways of refreshing the routing information from a neighbor:

- Clearing the neighbor relationship
- Soft-clearing the neighbor relationship (if soft reconfiguration is enabled for this specific neighbor)
- Using route refresh (if the neighbor supports this capability)

The **soft-reconfiguration inbound** feature consumes large volumes of memory in the Internet environment. The number of routes that can be received from a peer router on the Internet is so large that it is not feasible to store an extra copy.

The only reason for making the extra copy is to be able to replay the data through the new routing policy without tearing down the session and re-establishing it.

What is needed is a mechanism to ask the neighbor router to do a "clear soft outbound." If this were possible, the extra copy would not be needed. The neighboring router, of course, has its own copy in its BGP table, which it could resend to the local router whenever it is signaled to do so.

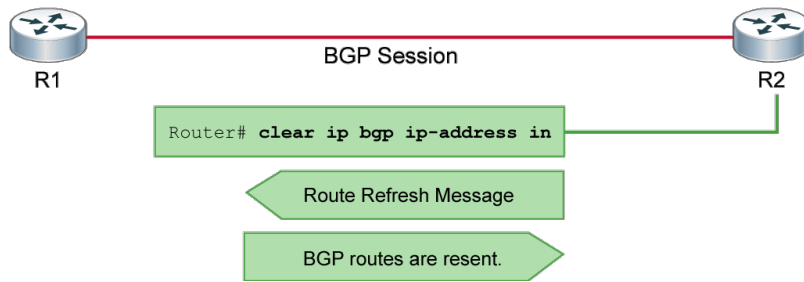
There is no such mechanism in standard BGP, but there is an optional BGP capability that allows one router to request a refresh from its neighbor: route refresh.

The table compares the various methods of BGP session reset, stating the advantages and disadvantages of each.



To use soft reset without preconfiguration, both BGP peers must support the soft route refresh capability, which is advertised in the Open message that is sent when the peers establish a [TCP](#) session. Routers that run Cisco IOS software releases earlier than Release 12.1 do not support the route refresh capability and must clear the BGP session using the **neighbor soft-reconfiguration** command.

Type of Reset	Advantages	Disadvantages
Hard reset	No memory overhead.	The prefixes in the BGP, IP, and FIB tables that the neighbor provides, are lost. Not recommended.
Outbound soft reset	No configuration, no storing of routing table updates.	Does not reset inbound routing table updates.
Dynamic inbound soft reset	Does not clear the BGP session or cache. Does not require storing of routing table updates, and has no memory overhead.	Both BGP routers must support the route refresh capability (Cisco IOS Software Release 12.1 and later releases).
Configured inbound soft reset (uses the neighbor soft-reconfiguration command)	Can be used when both BGP routers do not support the automatic route refresh capability.	Requires preconfiguration. Stores all received (inbound) routing policy updates without modification, and is thus memory-intensive. Recommended only when absolutely necessary.



- Step 1—Route refresh is negotiated when the BGP session is established.
- Step 2—Inbound routing policy is changed on R2.
- Step 3—Operator requests inbound route refresh.
- Step 4—R2 sends route refresh message to R1.
- Step 5—R1 resends all BGP routes to R2.

The router must negotiate the ability to use the route refresh feature when the BGP session is first established. The local router keeps a record that the capability is available with the neighbor. There is no need to keep a copy of the routing information that is received from the neighbor if it has the ability to refresh.

After reconfiguring the filters and route maps that will implement a new routing policy, you can issue the **clear ip bgp ip-address soft in** command in the local router. The router checks whether the route refresh capability is available, and if it is, requests resending of the BGP table of the neighbor instead of replaying its own copy.

Configuring Route Refresh

Use the **clear ip bgp * in** command, to send a route refresh message to all neighbors or **clear ip bgp ip-address in** to send a route refresh message to a specific neighbor.

```
router# clear ip bgp { * | ip-address | peer-group-name } in
```

- Sends a route refresh message to the neighbor or neighbors.
- Only works if the neighbor has previously advertised the route refresh capability.

You need not use the **soft** keyword, because soft reset is automatically assumed when the route refresh capability is supported.

To reset a [BGP](#) connection with BGP soft reconfiguration, use the **clear ip bgp** privileged EXEC command at the system prompt.

```
clear ip bgp { * | ip-address | peer-group-name } [soft [in | out]]
```

Syntax Description

Parameter	Description
*	Resets all current BGP sessions.
<i>ip-address</i>	Resets only the identified BGP neighbor.
<i>peer-group-name</i>	Resets the specified BGP peer group.
soft	(Optional) Soft reset. Does not reset the session
in out	(Optional) Triggers inbound or outbound soft reconfiguration. If the in or out option is not specified, both inbound and outbound soft reset are triggered.



When doing **clear ip bgp * in**, if **soft-reconfiguration inbound** is configured for a neighbor, it will cause a soft action. If **soft-reconfiguration inbound** is not configured, it will cause a route refresh.

Monitoring Route Refresh

You need to verify whether the [BGP](#) neighbor supports the route refresh capability.

```
R2# show ip bgp neighbors 172.16.22.22
BGP neighbor is 172.16.22.22, remote AS 200, external link
  BGP version 4, remote router ID 10.0.40.1
  BGP state = Established, up for 02:31:21
  Last read 00:00:23, last write 00:00:49, hold time is 180, keepalive interval is 60 seconds
  Neighbor sessions:
    1 active, is not multisession capable (disabled)
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Four-octets ASN Capability: advertised and received
    Address family IPv4 Unicast: advertised and received
    Enhanced Refresh Capability: advertised and received
```

- Verifies the support for route refresh capability.

Use the **show ip bgp neighbor** command to see whether the neighbor supports the route refresh message. In the example, neighbor 172.16.22.22 supports route refresh capability.

```
R2# debug ip bgp
*Mar 10 05:06:53.243: BGP: ses global 172.16.22.22 (0xECFD0708:0) act Send OPEN
*Mar 10 05:06:53.243: BGP: 172.16.22.22 active sending OPEN, version 4, my as: 1, holdtime 180 seconds, ID A000041
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active rcv message type 1, length (excl. header) 38
*Mar 10 05:06:53.247: BGP: ses global 172.16.22.22 (0xECFD0708:0) act Receive OPEN
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active rcv OPEN, version 4, holdtime 180 seconds
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active OPEN has CAPABILITY code: 128, length 0
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active OPEN has ROUTE-REFRESH capability(old) for all address-families
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active rcvd OPEN w/ optional parameter type 2 (Capability) len 2
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active OPEN has CAPABILITY code: 2, length 0
*Mar 10 05:06:53.247: BGP: 172.16.22.22 active OPEN has ROUTE-REFRESH capability(new) for all address-families
<... output omitted ...>
*Mar 10 05:06:53.248: BGP: 172.16.22.22 sending REFRESH_REQ(5) for afi/safi: 1/1, refresh code is 1
```

- Debug output after BGP session reset

Use **debug ip bgp** to display the negotiation of capabilities. Debugging displays the received capabilities.

The example shows a debugging output, after the BGP session was reset, using a hard reconfiguration. You can see, that a neighbor is advertising both old-style and standard (new-style) route refresh. The "old" refers to the old capability code of 128 that the old Cisco implementations of Route Refresh used, before the [RFC 2918](#) was in place. The "new" refers to the RFC 2918 compliant capability code 2 that all recent BGP implementations that support the Route Refresh capability, according to RFC 2918. After the session has been established, the router sends an initial standard route refresh message for the address family 1/1 ([IPv4](#) unicast).

```
R2# debug ip bgp
R2# debug ip bgp updates
R2# clear ip bgp 172.16.22.22 in
*Mar 10 05:34:28.458: BGP: 172.16.22.22 sending REFRESH_REQ(5) for afi/safi: 1/1, refresh code is 0
*Mar 10 05:34:28.459: BGP: 172.16.22.22 rcv message type 5, length (excl. header) 4
*Mar 10 05:34:28.459: BGP: 172.16.22.22 rcvd REFRESH_REQ for afi/safi: 1/1, refresh code is 1
*Mar 10 05:34:28.459: BGP: nbr_topo global 172.16.22.22 IPv4 Unicast:base (0xECFD0708:1) rcvd Refresh Start-of-RIB
*Mar 10 05:34:28.459: BGP: nbr_topo global 172.16.22.22 IPv4 Unicast:base (0xECFD0708:1) refresh_epoch is 3
*Mar 10 05:34:28.460: BGP(0): 172.16.22.22 rcvd UPDATE w/ attr: nexthop 17
```

```
2.16.22.22, origin i, metric 0, merged path 200, AS_PATH
*Mar 10 05:34:28.461: BGP(0): 172.16.22.22 rcvd 10.0.2.0/28
*Mar 10 05:34:28.461: BGP(0): 172.16.22.22 rcvd 10.0.2.16/28
*Mar 10 05:34:28.461: BGP(0): 172.16.22.22 rcvd 10.0.2.32/28
```

- Debug output after route refresh

Debugging also shows a route refresh message being sent to a neighbor after the network administrator issues the **clear ip bgp *ip-address* in** command from the privileged EXEC mode. When the neighbor 172.16.22.22 receives the route refresh message, it resends the BGP updates to the router R2. Updates are then processed against the configured BGP inbound filters on the R2 router.

Summary

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

- Because of the huge volumes of routing information, BGP cannot use traditional routing update methods
- The BGP Soft Reset Enhancement feature provides automatic support for dynamic soft reset of inbound BGP routing table updates that is not dependent upon stored routing table update information. This method requires no preconfiguration and needs much less memory than the previous soft reset method for inbound routing table updates.
- Route refresh is a BGP capability that is used to request a neighbor to resend routing information after configuration changes.
- The **clear ip bgp** ip-address in command sends a route refresh message to the neighboring router and executes if the neighbor has previously advertised the route refresh capability.
- To verify whether a neighbor supports route refresh, you can use the **show ip bgp neighbor** command. To display the negotiation process, you can use the **debug ip bgp** command.

Module Summary

Overview

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

- The multihomed customer network must exchange BGP information with both ISP networks. Dynamic routing is required for full redundancy, and BGP is the only protocol available that can be used in this scenario.
- An AS-path filter is created by an AS-path access-list. The access-list is applied to a set of routes from which a subset can be selected.
- Use prefix-lists to filter incoming or outgoing BGP updates to neighbors and to filter routes that are being redistributed into the BGP process from other routing protocols.
- Outbound route filtering is a mechanism that is used to minimize the number of updates that are requested from a neighbor.
- Route-maps provide a method to perform a variety of compound, complex filtering operations (such as dropping denied routes and modifying attributes of the permitted routes) within a single tool.
- Soft reconfiguration provides the ability to run all routes through filters without tearing down the sessions.

This module discussed BGP route filtering and BGP route selection policies. The module described multihomed BGP networks and identified the need for BGP route selection. This module also addressed configuring BGP to influence route selection by using AS-path filters, prefix-list filters, and route-maps. Outbound route filtering was also explained. In addition, details about soft reconfiguration and route refresh were provided.

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Sample Configuration for BGP with Two Different Service Providers (Multihoming)*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/23675-27.html>
- Cisco Systems, Inc. *Using Regular Expressions in BGP*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13754-26.html>
- Cisco Systems, Inc. *BGP Case Studies "BGP Case Studies 1."*
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html#BGPsec1>
- Cisco Systems, Inc. *BGP Case Studies "BGP Case Studies 3."*
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html#sec3>
- Cisco Systems, Inc. *Configuring BGP*.
http://www.cisco.com/c/en/us/td/docs/ios/12_2/ip/configuration/guide/fipr_c/1cfbgp.html
- Cisco Systems, Inc. *BGP Prefix-Based Outbound Route Filtering*.
http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-3e/irg-iproute-bgp-xe-3e-book/irg-oubound-route-filtering.pdf
- Cisco Systems, Inc. *Compatible Systems Setup Guides: BGP Configuration Guide*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/17612-bgp.html>
- Cisco Systems, Inc. *BGP Soft Reset Enhancement*.
http://www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/s_sfrst.html?mdfid=275960490

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. What are two reasons why a customer would want to connect to two ISPs? (Choose two.) (Source: "Using Multihomed BGP Networks")

- ☐ To expand capacity for Internet traffic.
- ☐ To better protect confidential information as it travels through the Internet.
- ☐ To provide redundancy to mission-critical services that are offered over the Internet.
- ☐ To efficiently route Internet traffic to two different divisions within the company.

1. What are the two technical requirements for multihomed customers? (Choose two.) (Source: "Using Multihomed BGP Networks")

☐ The ISPs must assign a range of IP network numbers to the customer.

☐ The customer network must exchange BGP information with each ISP network.

☐ In most cases, the customer must have its own public AS number.

☐ The customer network must not use AS path filters.

1. Which of the following statements best illustrates the importance of BGP policies that influence route selection in a multihomed BGP network? (Source: "Using Multihomed BGP Networks")

☐ The default BGP route selection does not always result in optimum routing.

☐ The default BGP route selection always results in optimum routing.

☐ After the route selection behavior has been set, it cannot be changed.

☐ The customer receives all routes from both service providers, giving redundancy; therefore, BGP policies are not necessary.

1. There is no need for ISPs and customers to influence the BGP path selection as BGP has built-in mechanisms to find the optimal way to reach the destination. True or false? (Source: "Using Multihomed BGP Networks")

☐ true

☐ false

1. Which three of the following are potential customer routing policies? (Choose three.) (Source: "Using Multihomed BGP Networks")

- ☐ One service provider is designated as primary, and the other is a backup.
- ☐ Traffic is load-balanced across both ISP networks.
- ☐ Traffic toward a specific destination goes through only one of the ISPs.
- ☐ Traffic to direct customers of the ISPs goes direct; all other traffic goes through the primary ISP.
- ☐ The two ISPs may have similar peering agreements with other ISPs.
- ☐ The ISPs use default routing to the customer, and the customer uses static routing to the ISP (or ISPs).

1. Which statement about the need to influence BGP route selection in a service provider environment is accurate? (Source: "Using Multihomed BGP Networks")

If both ISP connections terminate in one single customer router, only some routes that are received from the primary ISP can be assigned a BGP weight.

In most cases, it is more optimal to reach other customers connected to the backup ISP via the backup link, compared with reaching them via the primary link.

All routes that are received from the primary ISP over the primary link are assigned a local preference value, which is lower than the default value of 100.

When one ISP connection terminates in one single customer router, all routes that are received from that ISP are assigned a BGP weight.

1. Customer could become transit AS if no filtering to multihomed ISPs is performed. True or false? (Source: "Using Multihomed BGP Networks")

☒ true

☐ false

1. Which routes should customer announce to service providers in multihomed scenario? (Source: "Using Multihomed BGP Networks")

- Customer should announce only networks originating in customer's AS.
- Customer should announce all networks received from primary ISP to secondary ISP.
- Customer should announce all networks received from secondary ISP to primary ISP.
- Customer should not announce any networks.

1. ISP should allow customer to announce any possible networks via BGP. True or false? (Source: "Using Multihomed BGP Networks")

☐ true

☐ false

1. Which three goals represent appropriate reasons to apply AS path filters? (Choose three.) (Source: "Employing AS Path Filters")

- ☐ To ensure that only locally originated routes are announced.
- ☐ To limit routes that are advertised from IBGP neighbors.
- ☐ To select a subset of all routes based on their originating AS.
- ☐ To limit neighbor route updates to specific AS-originated routes.
- ☐ To ensure that all destination autonomous systems should be received from a specified neighbor.
- ☐ To change the weight or local preference attributes for all destination autonomous systems.

1. Which AS path is matched by the regular expression "72\$"?
(Source: "Employing AS Path Filters")

☐ 213 72 218 31 727

☐ 27 317 271 50 72

☐ 315 27 723 19 91

☐ 72 591 368 20 87

1. Which regular expression would you use to match all BGP networks originated in AS 89? (Source: "Employing AS Path Filters")

☐ ^89\$

☐ ^89_

☐ _89_

☐ _89\$

1. Which regular expression would you use to match all BGP networks going through AS 76? (Source: "Employing AS Path Filters")

☐ ^76\$

☐ ^76_

☐ _76_

☐ _76\$

1. Which BGP networks are matched by using regular expression ^54\$? (Source: "Employing AS Path Filters")

- ☐ All routes that are sourced in your directly connected neighboring AS 54.
- ☐ All routes that have AS 54 in their AS paths.
- ☐ All routes that are sourced in AS 54.
- ☐ All the routes that are reachable behind AS 54.

1. What is the difference between the regular expressions "_100_" and "_100\$"? (Source: "Employing AS Path Filters")

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.

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.

The first expression refers to routes that go through AS 100; the second expression refers to routes that originated in AS 100.

The first expression refers to routes that are directly connected to AS 100; the second expression refers to routes that originated in AS 100.

1. How do you match AS paths that contain exactly two single-digit AS numbers? (Source: "Employing AS Path Filters")

☐ use the expression "***"

☐ use the expression ".."

☐ use the expression "[0-9]_[0-9]"

☐ use the expression "^[0-9]_[0-9]\$" **Correct**

1. Which three steps are required to apply a new inbound routing policy to a neighbor? (Choose three.) (Source: "Employing AS Path Filters")

- ☐ Define an AS path access list.
- ☐ Attach the AS path filter to inbound or outbound updates for a specific BGP neighbor.
- ☐ Send incoming and outgoing AS path filters to the BGP neighbor.
- ☐ Force the updates to go through the new filter.

1. How can you test your regular expression? (Source: "Employing AS Path Filters")

- **show ip bgp access-list** command

- **show ip bgp filter** command

- **show ip bgp regexp** command

- **show ip bgp summary** command

1. What are two reasons that a multihomed customer needs prefix-lists? (Choose two.) (Source: "Filtering with Prefix-Lists")

- ☐ To ensure that only valid IP prefixes are announced to the ISPs.
- ☐ To set a limit on the number of prefixes that can be accepted from the ISPs.
- ☐ To prevent the customer from receiving its own IP prefixes from the ISP.
- ☐ To verify that the customer has received full Internet route tables.

1. Which three of the following choices are advantages of prefix lists over access lists? (Choose three.) (Source: "Filtering with Prefix-Lists")

- ☐ significant performance improvement on long filters
- ☐ support for incremental updates
- ☐ ability to consist of any number of lines, each of which indicates a test and a result
- ☐ flexibility
- ☐ more complex command-line interface
- ☐ sequential scanning of prefix-lists within Cisco IOS software

1. When you define prefix lists, what are two reasons to use sequence numbers? (Choose two.) (Source: "Filtering with Prefix-Lists")

- ☐ To reference the associated access list for the prefix list entry.
- ☐ To provide a means of linking an AS path filter list to the prefix list.
- ☐ To provide an execution order for prefix-list entries.
- ☐ To provide a means of inserting or deleting list entries.

1. Which of the following statements is accurate about the **ge** and **le** parameters in the **ip prefix-list** global configuration command required to configure prefix-list filters? (Source: "Filtering with Prefix-Lists")

☐ No match is assumed when neither **ge** nor **le** is specified.

☐ The range is assumed to be from *ge-value* to 142 only if the **ge** attribute is specified.

☐ The range is assumed to be from *len* to 32 only if the **le** attribute is specified.

☐ An exact match is assumed when neither **ge** nor **le** is specified.

1. Which of the following routes will be matched by the the prefix list **10.0.1.0/24 ge 28**? (Source: "Filtering with Prefix Lists")

☐ 10.0.1.0/26

☐ 10.1.1.0/28

☐ 10.0.1.0/29

☐ 10.0.0.0/28

1. Which of the following statements about implementing prefix lists in a BGP network is accurate? (Source: "Filtering with Prefix Lists")

☐ You can optionally apply filter lists and prefix lists on either incoming or outgoing neighbors in any combination.

☐ You can optionally apply filter lists and prefix lists only on outgoing neighbors in any combination

☐ Either the incoming prefix list or the incoming filter list must permit the routes that are received from a neighbor before they are accepted into the BGP table.

☐ Outgoing routes must pass the outgoing prefix list before being transmitted to the neighbor.

1. Which three steps are required to apply a new inbound routing policy (based on prefix list) to a neighbor? (Choose three.) (Source: "Filtering with Prefix Lists")

- ☐ Define an IP prefix list.
- ☐ Attach the prefix list to inbound or outbound updates for a specific BGP neighbor.
- ☐ Send incoming and outgoing AS path filters to the BGP neighbor.
- ☐ Force the updates to go through the new filter.

1. How can you apply the same prefix list to multiple BGP neighbors on a router? (Source: "Filtering with Prefix Lists")

- By configuring a neighbor prefix-list statement for each BGP peer.

- By configuring a neighbor distribute-list statement for each neighbor.

- By using the BGP *peer-group* option with the **neighbor** statement.

- By configuring the prefix list as a global filter under the BGP routing process.

1. How can you use the **show ip prefix-list** command to display the prefix list entry that matches a specific prefix and length? (Source: "Filtering with Prefix Lists")

not a feature of the **show ip prefix-list** command

by specifying the **detail** keyword

with the **longer** keyword to display all matches except those with more specific entries

by specifying the **first-match** keyword

1. Which of the following best describes the capabilities of the proprietary ORF type that is supported on Cisco routers? (Source: "Using Outbound Route Filtering")

☐ standard BGP communities filtering

☐ extended BGP communities filtering

☐ AS path filtering

☐ prefix list filtering

1. ORF feature is more effective than standard inbound filtering when it comes to the amount of resources that are required for routing update generation and processing. True or false? (Source: "Using Outbound Route Filtering")

☐ true

☐ false

1. What are two key benefits to using outbound route filtering? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☐ conserves CPU cycles
- ☐ improves security
- ☐ reduces bandwidth that is used by unnecessary routing updates
- ☐ increases neighbor availability

1. Which two of the following statements about BGP prefix-based outbound route filtering are accurate? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☐ Uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers.
- ☐ Can limit the number of unwanted routing updates.
- ☐ Increases the amount of resources required to receive and discard routes that would otherwise be filtered out.
- ☐ Can be used to increase the amount of processing on a router that is not accepting full routes from a service provider network.

1. How should you configure the **neighbor capability orf prefix-list** command on a router that is applying a prefix list filter as an outbound route policy? (Source: "Using Outbound Route Filtering")

☐ send

☐ receive

☐ both

☐ prefix filter

1. Which of the following is true regarding ORF capabilities options? (Choose two.) (Source: "Using Outbound Route Filtering")

☐ **"both"** option allows sending and receiving of prefix lists.

☐ **"send"** option allows receiving of prefix lists.

☐ **"receive"** option allows receiving of prefix lists.

☐ **"both"** option allows only sending of prefix lists.

1. What are two methods of determining that a router has ORF capabilities exchange configured? (Choose two.) (Source: "Using Outbound Route Filtering")

☐ **show running-config | begin bgp** command

☐ **show ip bgp negotiate** command

☐ **show ip bgp neighbors** command

☐ **show ip prefix-list** command

1. Which command should be used to verify that IP prefix-list was received on the ORF receiver? (Source: "Using Outbound Route Filtering")

☐ **show ip bgp neighbors *IP_address* received prefix-filter**

☐ **show ip prefix-list**

☐ **show ip bgp neighbors *IP_address* send prefix-filter**

☐ **show orf prefix-list received**

1. What are two prerequisites before you can configure ORF prefix list functionality? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☐ A route refresh must be sent using the **clear ip bgp** command.
- ☐ A BGP peering session between the ORF routers must be up and running.
- ☐ ORF capabilities must be enabled on both routers.
- ☐ You must configure a prefix list filter on the receiving router.

1. Which of the following statements about the function of a route map is accurate? (Source: "Applying Route Maps as BGP Filters")

☐ A route map cannot be used to modify attributes of the permitted routes.

☐ A route map is a filter that uses a series of match conditions, and that which is denied by the route map is dropped.

☐ A route map is less complex than the access list.

☐ Each route map statement starts with a series of match clauses.

1. Which three of the following statements are accurate about the BGP Route Map Policy List Support feature? (Choose three.) (Source: "Applying Route Maps as BGP Filters")

☐ The BGP Route Map Policy List Support feature allows a network operator to group route-map **match** clauses into named lists called policy-lists.

☐ The network operator manually reconfigures each recurring group of match clauses that occur in multiple route-map entries.

☐ The AND and OR semantics in the route map function differently for policy-lists than for **match** and **set** clauses.

☐ To create a BGP policy-list, use the **ip policy-list** command in policy-map configuration mode.

☐ To configure a route-map to evaluate and process a BGP policy-list in a route map, use the **match policy-list** command in route-map configuration mode.

☐ To display information about a configured policy-list and policy-list entries, use the **show ip policy-list** command in route-map configuration mode.

1. Which two of the following are functions of the BGP Route Map Continue feature? (Choose two.) (Source: "Applying Route Maps as BGP Filters")

- ☐ Provides the ability to pause if a sequence number is not specified.
- ☐ Provides the capability to execute additional entries in a route-map after an entry is executed with successful **match** and **set** clauses.
- ☐ Allows modularization of network policy configuration so that repeated policy definitions can be expanded within the same route map.
- ☐ Allows configuration and organization of more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map.

1. Which of the following commands is used to distribute any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or to perform policy routing on packets? (Source: "Applying Route Maps as BGP Filters")

☐ **match ip next-hop**

☐ **match ip route-source**

☐ **match ip address**

☐ **show ip bgp route-map**

1. How do you implement a "**permit all**" statement when you are using route maps? (Source: "Applying Route Maps as BGP Filters")

- By default, a route-map has an "implicit permit any" statement if no match is found.

- You must configure a route map with a "**permit**" parameter and no match clause.

- You must configure a route map with a "**deny**" parameter and a "deny none" clause.

- You must configure a route map with a "**permit any**" match clause.

1. What happens to incoming BGP updates that do not match any route-map match clauses? (Source: "Applying Route Maps as BGP Filters")

• They are entered into the BGP table.

• They are entered into the BGP table and marked with a weight of 32768.

• They are not accepted by the router or entered into the BGP table.

• They are entered into the BGP table if a matching route exists in the IP routing table.

1. Which three BGP attributes can you set using route maps? (Choose three.) (Source: "Applying Route Maps as BGP Filters")

☐	MED
☐	path origin
☐	administrative distance
☐	weight metric
☐	next-hop
☐	atomic aggregate

1. What are two reasons for using route map sequence numbers? (Choose two.) (Source: "Applying Route Maps as BGP Filters")

- ☐ To allow insertion or deletion of route-map entries.
- ☐ To order the execution sequence of route-map match clauses.
- ☐ To provide an ordered execution sequence for the route map.
- ☐ To map between prefix-list statements and route-map match clauses.

1. Which command should be used on R1 (IP address 1.1.1.1) to apply a route map named Filter to incoming BGP updates coming from R2 (IP address 2.2.2.2)? (Source: "Applying Route Maps as BGP Filters")

☐ neighbor 2.2.2.2 route-map Filter in

☐ neighbor 1.1.1.1 route-map Filter in

☐ neighbor 2.2.2.2 route-map Filter out

☐ neighbor 1.1.1.1 route-map Filter out

1. Why is clearing a BGP session a disruptive change in routing policy?
(Source: "Implementing Changes in BGP Policy")

• Clearing a BGP session takes a long time and can disrupt packet forwarding.

• You cannot recover information that is sent while the BGP session is being cleared.

• You cannot automatically re-establish sessions that are torn down during the clearing operation.

• You cannot selectively tear down BGP sessions; you must clear sessions with all neighbors.

1. Using the **clear ip bgp *** command, all BGP sessions will stay up and no BGP routes are lost. True or false? (Source: "Implementing Changes in BGP Policy")

☐ true

☐ false

1. Which two of the following are functions of the BGP Soft Reset Enhancement feature? (Choose two.) (Source: "Implementing Changes in BGP Policy")

- ☐ allows dynamic route refresh requests
- ☐ requires no preconfiguration
- ☐ provides newer method for inbound soft reset that uses stored inbound routing table updates
- ☐ uses expanded memory

1. BGP route refresh is BGP capability that allows a BGP router to request retransmission of BGP routes from a neighbor. True or false? (Source: "Implementing Changes in BGP Policy")

☐ true

☐ false

1. Which of the following statements about the command that is required to perform a route refresh is accurate? (Source: "Implementing Changes in BGP Policy")

☐ You will use the **clear ip bgp * in** command to send a route refresh message to all neighbors.

☐ You will use the **clear ip bgp *ip-address* in** command to send a route refresh message to all neighbors.

☐ You must use the **soft** keyword with the **clear ip bgp** command because soft reset is not automatically assumed when the route refresh capability is supported.

☐ The **clear ip bgp** command works even if the neighbor has not previously advertised the route refresh capability.

1. Which command would you use on R1 (IP address 1.1.1.1) to request from neighboring R2 (IP address 2.2.2.2) to resend routing information? (Source: "Implementing Changes in BGP Policy")

☐ **clear ip bgp 1.1.1.1 out**

☐ **clear ip bgp 1.1.1.1 in**

☐ **clear ip bgp 2.2.2.2 out**

☐ **clear ip bgp 2.2.2.2 in**

1. Which RFC 2918 compliant capability code 2 used by all recent BGP implementations will tell you that router supports the Route Refresh feature? (Source: "Implementing Changes in BGP Policy")

1

218

4

2

1. Which command should be used to display the negotiation of BGP capabilities after session reset? (Source: "Implementing Changes in BGP Policy")

☐ **debug ip**

☐ **debug bgp packets**

☐ **debug ip bgp updates**

☐ **debug ip bgp**

1. How do you determine whether a BGP neighbor supports route refresh? (Source: "Implementing Changes in BGP Policy")

- A flag in the BGP table indicates the presence of route refresh capability

- The **show ip bgp neighbor** command indicates whether the option is supported.

- Initiate the **debug ip bgp negotiation** command to see whether the router has completed a route refresh capabilities exchange.

- Execute the **clear ip bgp *** command. Command-line BGP status messages will indicate route refresh support capabilities.

Answer Key

1. What are two reasons why a customer would want to connect to two ISPs? (Choose two.) (Source: "Using Multihomed BGP Networks")

- ☒ To expand capacity for Internet traffic.
- ☐ To better protect confidential information as it travels through the Internet.
- ☒ To provide redundancy to mission-critical services that are offered over the Internet.
- ☐ To efficiently route Internet traffic to two different divisions within the company.

1. What are the two technical requirements for multihomed customers? (Choose two.) (Source: "Using Multihomed BGP Networks")

☐ The ISPs must assign a range of IP network numbers to the customer.

☒ The customer network must exchange BGP information with each ISP network.

☒ In most cases, the customer must have its own public AS number.

☐ The customer network must not use AS path filters.

1. Which of the following statements best illustrates the importance of BGP policies that influence route selection in a multihomed BGP network? (Source: "Using Multihomed BGP Networks")

☐ The default BGP route selection does not always result in optimum routing.

☐ The default BGP route selection always results in optimum routing.

☐ After the route selection behavior has been set, it cannot be changed.

☐ The customer receives all routes from both service providers, giving redundancy; therefore, BGP policies are not necessary.

1. There is no need for ISPs and customers to influence the BGP path selection as BGP has built-in mechanisms to find the optimal way to reach the destination. True or false? (Source: "Using Multihomed BGP Networks")

☐ true

☐ false

1. Which three of the following are potential customer routing policies? (Choose three.) (Source: "Using Multihomed BGP Networks")

- ☒ One service provider is designated as primary, and the other is a backup.
- ☐ Traffic is load-balanced across both ISP networks.
- ☒ Traffic toward a specific destination goes through only one of the ISPs.
- ☒ Traffic to direct customers of the ISPs goes direct; all other traffic goes through the primary ISP.
- ☐ The two ISPs may have similar peering agreements with other ISPs.
- ☐ The ISPs use default routing to the customer, and the customer uses static routing to the ISP (or ISPs).

1. Which statement about the need to influence BGP route selection in a service provider environment is accurate? (Source: "Using Multihomed BGP Networks")

If both ISP connections terminate in one single customer router, only some routes that are received from the primary ISP can be assigned a BGP weight.

In most cases, it is more optimal to reach other customers connected to the backup ISP via the backup link, compared with reaching them via the primary link.

All routes that are received from the primary ISP over the primary link are assigned a local preference value, which is lower than the default value of 100.

When one ISP connection terminates in one single customer router, all routes that are received from that ISP are assigned a BGP weight.

1. Customer could become transit AS if no filtering to multihomed ISPs is performed. True or false? (Source: "Using Multihomed BGP Networks")

☒ true

☐ false

1. Which routes should customer announce to service providers in multihomed scenario? (Source: "Using Multihomed BGP Networks")

- ☐ Customer should announce only networks originating in customer's AS.
- ☐ Customer should announce all networks received from primary ISP to secondary ISP.
- ☐ Customer should announce all networks received from secondary ISP to primary ISP.
- ☐ Customer should not announce any networks.

1. ISP should allow customer to announce any possible networks via BGP. True or false? (Source: "Using Multihomed BGP Networks")

☐ true

☐ false

1. Which three goals represent appropriate reasons to apply AS path filters? (Choose three.) (Source: "Employing AS Path Filters")



To ensure that only locally originated routes are announced.



To limit routes that are advertised from IBGP neighbors.



To select a subset of all routes based on their originating AS.



To limit neighbor route updates to specific AS-originated routes.



To ensure that all destination autonomous systems should be received from a specified neighbor.



To change the weight or local preference attributes for all destination autonomous systems.

1. Which AS path is matched by the regular expression "72\$"?
(Source: "Employing AS Path Filters")

213 72 218 31 727

27 317 271 50 72

315 27 723 19 91

72 591 368 20 87

1. Which regular expression would you use to match all BGP networks originated in AS 89? (Source: "Employing AS Path Filters")

☐ ^89\$

☐ ^89_

☐ _89_

☐ _89\$

1. Which regular expression would you use to match all BGP networks going through AS 76? (Source: "Employing AS Path Filters")

☐ ^76\$

☐ ^76_

☐ _76_

☐ _76\$

1. Which BGP networks are matched by using regular expression ^54\$? (Source: "Employing AS Path Filters")

- ☐ All routes that are sourced in your directly connected neighboring AS 54.
- ☐ All routes that have AS 54 in their AS paths.
- ☐ All routes that are sourced in AS 54.
- ☐ All the routes that are reachable behind AS 54.

1. What is the difference between the regular expressions "_100_" and "_100\$"? (Source: "Employing AS Path Filters")

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.

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.

The first expression refers to routes that go through AS 100; the second expression refers to routes that originated in AS 100.

The first expression refers to routes that are directly connected to AS 100; the second expression refers to routes that originated in AS 100.

1. How do you match AS paths that contain exactly two single-digit AS numbers? (Source: "Employing AS Path Filters")

use the expression "***"

use the expression ".."

use the expression "[0-9]_[0-9]"

use the expression "^[0-9]_[0-9]\$"

1. Which three steps are required to apply a new inbound routing policy to a neighbor? (Choose three.) (Source: "Employing AS Path Filters")

- ☒ Define an AS path access list.
- ☒ Attach the AS path filter to inbound or outbound updates for a specific BGP neighbor.
- ☐ Send incoming and outgoing AS path filters to the BGP neighbor.
- ☒ Force the updates to go through the new filter.

1. How can you test your regular expression? (Source: "Employing AS Path Filters")

 **show ip bgp access-list** command

 **show ip bgp filter** command

 **show ip bgp regexp** command

 **show ip bgp summary** command

1. What are two reasons that a multihomed customer needs prefix-lists? (Choose two.) (Source: "Filtering with Prefix-Lists")

- ☒ To ensure that only valid IP prefixes are announced to the ISPs.
- ☐ To set a limit on the number of prefixes that can be accepted from the ISPs.
- ☒ To prevent the customer from receiving its own IP prefixes from the ISP.
- ☐ To verify that the customer has received full Internet route tables.

1. Which three of the following choices are advantages of prefix lists over access lists? (Choose three.) (Source: "Filtering with Prefix-Lists")

- ☒ significant performance improvement on long filters
- ☒ support for incremental updates
- ☐ ability to consist of any number of lines, each of which indicates a test and a result
- ☒ flexibility
- ☐ more complex command-line interface
- ☐ sequential scanning of prefix-lists within Cisco IOS software

1. When you define prefix lists, what are two reasons to use sequence numbers? (Choose two.) (Source: "Filtering with Prefix-Lists")

- ☐ To reference the associated access list for the prefix list entry.
- ☐ To provide a means of linking an AS path filter list to the prefix list.
- ☒ To provide an execution order for prefix-list entries.
- ☒ To provide a means of inserting or deleting list entries.

1. Which of the following statements is accurate about the **ge** and **le** parameters in the **ip prefix-list** global configuration command required to configure prefix-list filters? (Source: "Filtering with Prefix-Lists")

☐ No match is assumed when neither **ge** nor **le** is specified.

☐ The range is assumed to be from *ge-value* to 142 only if the **ge** attribute is specified.

☐ The range is assumed to be from *len* to 32 only if the **le** attribute is specified.

☐ An exact match is assumed when neither **ge** nor **le** is specified.

1. Which of the following routes will be matched by the the prefix list **10.0.1.0/24 ge 28**? (Source: "Filtering with Prefix Lists")

☐ 10.0.1.0/26

☐ 10.1.1.0/28

☐ 10.0.1.0/29

☐ 10.0.0.0/28

1. Which of the following statements about implementing prefix lists in a BGP network is accurate? (Source: "Filtering with Prefix Lists")

- ☐ You can optionally apply filter lists and prefix lists on either incoming or outgoing neighbors in any combination.
- ☐ You can optionally apply filter lists and prefix lists only on outgoing neighbors in any combination
- ☐ Either the incoming prefix list or the incoming filter list must permit the routes that are received from a neighbor before they are accepted into the BGP table.
- ☐ Outgoing routes must pass the outgoing prefix list before being transmitted to the neighbor.

1. Which three steps are required to apply a new inbound routing policy (based on prefix list) to a neighbor? (Choose three.) (Source: "Filtering with Prefix Lists")

- ☒ Define an IP prefix list.
- ☒ Attach the prefix list to inbound or outbound updates for a specific BGP neighbor.
- ☐ Send incoming and outgoing AS path filters to the BGP neighbor.
- ☒ Force the updates to go through the new filter.

1. How can you apply the same prefix list to multiple BGP neighbors on a router? (Source: "Filtering with Prefix Lists")

By configuring a neighbor prefix-list statement for each BGP peer.

By configuring a neighbor distribute-list statement for each neighbor.

By using the BGP *peer-group* option with the **neighbor** statement.

By configuring the prefix list as a global filter under the BGP routing process.

1. How can you use the **show ip prefix-list** command to display the prefix list entry that matches a specific prefix and length? (Source: "Filtering with Prefix Lists")

☐ not a feature of the **show ip prefix-list** command

☐ by specifying the **detail** keyword

☐ with the **longer** keyword to display all matches except those with more specific entries

☐ by specifying the **first-match** keyword

1. Which of the following best describes the capabilities of the proprietary ORF type that is supported on Cisco routers? (Source: "Using Outbound Route Filtering")

☐ standard BGP communities filtering

☐ extended BGP communities filtering

☐ AS path filtering

☐ prefix list filtering

1. ORF feature is more effective than standard inbound filtering when it comes to the amount of resources that are required for routing update generation and processing. True or false? (Source: "Using Outbound Route Filtering")

☐ true

☐ false

1. What are two key benefits to using outbound route filtering? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☒ conserves CPU cycles
- ☐ improves security
- ☒ reduces bandwidth that is used by unnecessary routing updates
- ☐ increases neighbor availability

1. Which two of the following statements about BGP prefix-based outbound route filtering are accurate? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☒ Uses BGP ORF send and receive capabilities to minimize the number of BGP updates that are sent between BGP peers.
- ☒ Can limit the number of unwanted routing updates.
- ☐ Increases the amount of resources required to receive and discard routes that would otherwise be filtered out.
- ☐ Can be used to increase the amount of processing on a router that is not accepting full routes from a service provider network.

1. How should you configure the **neighbor capability orf prefix-list** command on a router that is applying a prefix list filter as an outbound route policy? (Source: "Using Outbound Route Filtering")

☐ send

☐ receive

☐ both

☐ prefix filter

1. Which of the following is true regarding ORF capabilities options? (Choose two.) (Source: "Using Outbound Route Filtering")

☒ **"both"** option allows sending and receiving of prefix lists.

☐ **"send"** option allows receiving of prefix lists.

☒ **"receive"** option allows receiving of prefix lists.

☐ **"both"** option allows only sending of prefix lists.

1. What are two methods of determining that a router has ORF capabilities exchange configured? (Choose two.) (Source: "Using Outbound Route Filtering")

☒ **show running-config | begin bgp** command

☐ **show ip bgp negotiate** command

☒ **show ip bgp neighbors** command

☐ **show ip prefix-list** command

1. Which command should be used to verify that IP prefix-list was received on the ORF receiver? (Source: "Using Outbound Route Filtering")

show ip bgp neighbors *IP_address* received prefix-filter

show ip prefix-list

show ip bgp neighbors *IP_address* send prefix-filter

show orf prefix-list received

1. What are two prerequisites before you can configure ORF prefix list functionality? (Choose two.) (Source: "Using Outbound Route Filtering")

- ☐ A route refresh must be sent using the **clear ip bgp** command.
- ☒ A BGP peering session between the ORF routers must be up and running.
- ☒ ORF capabilities must be enabled on both routers.
- ☐ You must configure a prefix list filter on the receiving router.

1. Which of the following statements about the function of a route map is accurate? (Source: "Applying Route Maps as BGP Filters")

☐ A route map cannot be used to modify attributes of the permitted routes.

☐ A route map is a filter that uses a series of match conditions, and that which is denied by the route map is dropped.

☐ A route map is less complex than the access list.

☐ Each route map statement starts with a series of match clauses.

1. Which three of the following statements are accurate about the BGP Route Map Policy List Support feature? (Choose three.) (Source: "Applying Route Maps as BGP Filters")

☒ The BGP Route Map Policy List Support feature allows a network operator to group route-map **match** clauses into named lists called policy-lists.

☐ The network operator manually reconfigures each recurring group of match clauses that occur in multiple route-map entries.

☐ The AND and OR semantics in the route map function differently for policy-lists than for **match** and **set** clauses.

☒ To create a BGP policy-list, use the **ip policy-list** command in policy-map configuration mode.

☒ To configure a route-map to evaluate and process a BGP policy-list in a route map, use the **match policy-list** command in route-map configuration mode.

☐ To display information about a configured policy-list and policy-list entries, use the **show ip policy-list** command in route-map configuration mode.

1. Which two of the following are functions of the BGP Route Map Continue feature? (Choose two.) (Source: "Applying Route Maps as BGP Filters")

- ☐ Provides the ability to pause if a sequence number is not specified.
- ☒ Provides the capability to execute additional entries in a route-map after an entry is executed with successful **match** and **set** clauses.
- ☐ Allows modularization of network policy configuration so that repeated policy definitions can be expanded within the same route map.
- ☒ Allows configuration and organization of more modular policy definitions to reduce the number of policy configurations that are repeated within the same route map.

1. Which of the following commands is used to distribute any routes that have a destination network number address that is permitted by a standard access list, an extended access list, or a prefix list, or to perform policy routing on packets? (Source: "Applying Route Maps as BGP Filters")

☐ **match ip next-hop**

☐ **match ip route-source**

☐ **match ip address**

☐ **show ip bgp route-map**

1. How do you implement a "**permit all**" statement when you are using route maps? (Source: "Applying Route Maps as BGP Filters")

By default, a route-map has an "implicit permit any" statement if no match is found.

You must configure a route map with a "**permit**" parameter and no match clause.

You must configure a route map with a "**deny**" parameter and a "deny none" clause.

You must configure a route map with a "**permit any**" match clause.

1. What happens to incoming BGP updates that do not match any route-map match clauses? (Source: "Applying Route Maps as BGP Filters")

They are entered into the BGP table.

They are entered into the BGP table and marked with a weight of 32768.

They are not accepted by the router or entered into the BGP table.

They are entered into the BGP table if a matching route exists in the IP routing table.

1. Which three BGP attributes can you set using route maps? (Choose three.) (Source: "Applying Route Maps as BGP Filters")

☒	MED
☐	path origin
☐	administrative distance
☒	weight metric
☒	next-hop
☐	atomic aggregate

1. What are two reasons for using route map sequence numbers? (Choose two.) (Source: "Applying Route Maps as BGP Filters")

- ☒ To allow insertion or deletion of route-map entries.
- ☐ To order the execution sequence of route-map match clauses.
- ☒ To provide an ordered execution sequence for the route map.
- ☐ To map between prefix-list statements and route-map match clauses.

1. Which command should be used on R1 (IP address 1.1.1.1) to apply a route map named Filter to incoming BGP updates coming from R2 (IP address 2.2.2.2)? (Source: "Applying Route Maps as BGP Filters")

☐ neighbor 2.2.2.2 route-map Filter in

☐ neighbor 1.1.1.1 route-map Filter in

☐ neighbor 2.2.2.2 route-map Filter out

☐ neighbor 1.1.1.1 route-map Filter out

1. Why is clearing a BGP session a disruptive change in routing policy?
(Source: "Implementing Changes in BGP Policy")

Clearing a BGP session takes a long time and can disrupt packet forwarding.

You cannot recover information that is sent while the BGP session is being cleared.

You cannot automatically re-establish sessions that are torn down during the clearing operation.

You cannot selectively tear down BGP sessions; you must clear sessions with all neighbors.

1. Using the **clear ip bgp *** command, all BGP sessions will stay up and no BGP routes are lost. True or false? (Source: "Implementing Changes in BGP Policy")

☒ true

☐ false

1. Which two of the following are functions of the BGP Soft Reset Enhancement feature? (Choose two.) (Source: "Implementing Changes in BGP Policy")

- ☒ allows dynamic route refresh requests
- ☒ requires no preconfiguration
- ☐ provides newer method for inbound soft reset that uses stored inbound routing table updates
- ☐ uses expanded memory

1. BGP route refresh is BGP capability that allows a BGP router to request retransmission of BGP routes from a neighbor. True or false? (Source: "Implementing Changes in BGP Policy")

☒ true

☐ false

1. Which of the following statements about the command that is required to perform a route refresh is accurate? (Source: "Implementing Changes in BGP Policy")

☐ You will use the **clear ip bgp * in** command to send a route refresh message to all neighbors.

☐ You will use the **clear ip bgp *ip-address* in** command to send a route refresh message to all neighbors.

☐ You must use the **soft** keyword with the **clear ip bgp** command because soft reset is not automatically assumed when the route refresh capability is supported.

☐ The **clear ip bgp** command works even if the neighbor has not previously advertised the route refresh capability.

1. Which command would you use on R1 (IP address 1.1.1.1) to request from neighboring R2 (IP address 2.2.2.2) to resend routing information? (Source: "Implementing Changes in BGP Policy")

☐ **clear ip bgp 1.1.1.1 out**

☐ **clear ip bgp 1.1.1.1 in**

☐ **clear ip bgp 2.2.2.2 out**

☐ **clear ip bgp 2.2.2.2 in**

1. Which RFC 2918 compliant capability code 2 used by all recent BGP implementations will tell you that router supports the Route Refresh feature? (Source: "Implementing Changes in BGP Policy")

1

218

4

2

1. Which command should be used to display the negotiation of BGP capabilities after session reset? (Source: "Implementing Changes in BGP Policy")

☐ **debug ip**

☐ **debug bgp packets**

☐ **debug ip bgp updates**

☒ **debug ip bgp**

1. How do you determine whether a BGP neighbor supports route refresh? (Source: "Implementing Changes in BGP Policy")

A flag in the BGP table indicates the presence of route refresh capability

The **show ip bgp neighbor** command indicates whether the option is supported.

Initiate the **debug ip bgp negotiation** command to see whether the router has completed a route refresh capabilities exchange.

Execute the **clear ip bgp *** command. Command-line BGP status messages will indicate route refresh support capabilities.

Route Selection Using Attributes

Introduction

Routes that are learned via [BGP](#) have properties that are associated with them. They aid a router in determining the best route to a destination when multiple paths to that particular destination exist. These properties are referred to as BGP attributes. In this module, you will learn the role of BGP attributes, and how their presence influences route selection in BGP.

Understanding how BGP attributes, including weight, local preference, [AS](#) path prepending, [MED](#), and BGP communities, influence route selection is required for the design of robust networks.

Upon completing this module, you will be able to:

- Define how to use BGP weight attribute to influence the route selection
- Define how to use BGP local preference attribute to influence the route selection
- Define how to use BGP AS path prepending to influence the return path that is selected by the neighboring autonomous systems
- Define how to use MED attribute to influence the route selection
- Define how to use BGP community attributes to influence the route selection

Influencing BGP Route Selection with Weights

Overview

When connections to multiple providers are required, it is important that [BGP](#) selects the optimum route for traffic to use. The optimum, or best, route may not be what you intended during the design, based on the design criteria, administrative policies, or corporate mandate. Fortunately, BGP provides many tools for you to influence route selection. One of these tools is the weight attribute.

In this lesson, you will learn how to influence BGP route selection by setting the weight attribute of incoming BGP routes. Two methods that are used to set the weight attribute, default weight and route maps, are discussed in this lesson. You will also learn how to monitor the BGP table to verify correct weight configuration and properly influence path selection.

Upon completing this lesson, you will be able to:

- List BGP route selection criteria for best-path route selection
- Describe the use of BGP weights to influence the BGP route selection process
- Identify the Cisco IOS commands that are required to configure per-neighbor weights
- Influence the BGP route selection process by configuring per-neighbor weights
- Describe how to influence the BGP route selection process by configuring BGP weights with route maps
- Summarize BGP route selection and filtering tools

BGP Route Selection Criteria

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, the next criterion, local preference, is checked.

- Prefer highest weight (local to router)
- Prefer highest local preference (global within AS)
- Prefer routes that the router originated
- Prefer shorter AS paths (only length is compared)
- Prefer lowest origin code (IGP < EGP < Incomplete)
- Prefer lowest MED
- Prefer external (EBGP) paths over internal (IBGP)
- For IBGP paths, prefer path through closest IGP neighbor
- For EBGP paths, prefer oldest (most stable) path
- Prefer paths from router with the lower BGP router-ID

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

Influencing BGP Route Selection

The weight attribute is local to a single router only. BGP protocol never propagates the weight value, and this value constitutes a routing policy local to the router. You have two different ways how to assign the weight attribute to a route.

- 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
 - Complex criteria with route maps

The weight attribute can be assigned to a route using one of the following:

- Assign a default weight value to all of the routes that are received from a specific neighbor. This weight value indicates that the neighbor is preferred over the other neighbors.
- Apply a route map on incoming routes from a neighbor to select some routes and assign weight values to them. Remember that a route map also acts as a filter and will silently drop the routes that the route map does not permit in any statement.

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.

When a route map is applied, it is configured on the router. The route map can be arbitrarily complex and can use various selection criteria, such as a network number or AS path to select routes. You can alter some attributes of the selected routes. The route map can set the weight values of permitted routes. Selection can be done in several route map statements, giving the opportunity to assign a certain weight value to some routes and another weight value to others. A route map can also completely filter out routes.



If you want the routing policy to be exchanged within all the routers in the [AS](#), assign *local preference* attribute to a route. 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.

Configuring Per-Neighbor Weights

If you want all routes from the [BGP](#) neighbor to get the specified weight, you have to configure per-neighbor weights.

```
router(config-router)# neighbor ip-address weight weight
```

- All routes from the BGP neighbor get the specified weight.
- BGP routes with higher weight are preferred.
- Weight is applied only to new incoming updates.
- To enforce new weights, reestablish BGP sessions with your neighbors by using the **clear ip bgp** command.

To assign a weight to a neighbor connection, use the **neighbor weight** router configuration command. To remove a weight assignment, use the **no** form of this command.

All routes that a router receives from the neighbor after the configuration line is in place get the weight value assigned. To make sure that all routes from the neighbor receive the new weight value, you can restart the BGP session using **clear ip bgp** command, forcing the neighbor to resend all routes.

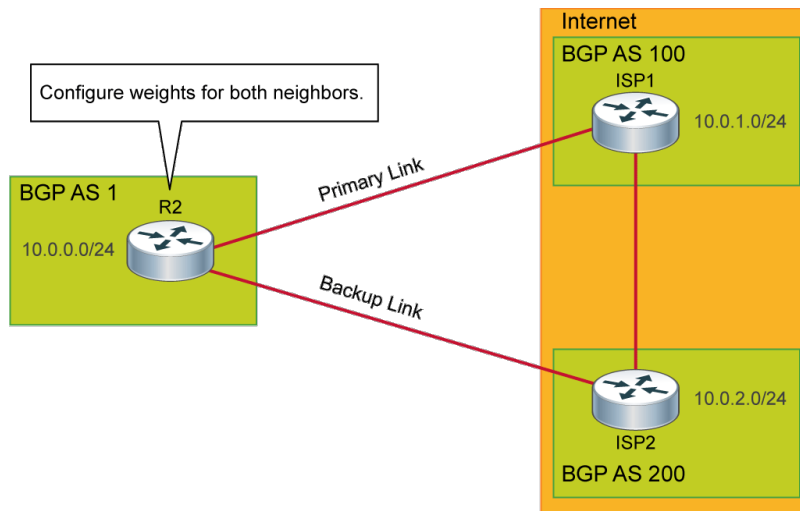


If no weight value is specified, the default value for routes learned through another BGP peer is 0 and the default value for routes sourced by the local router is 32768.

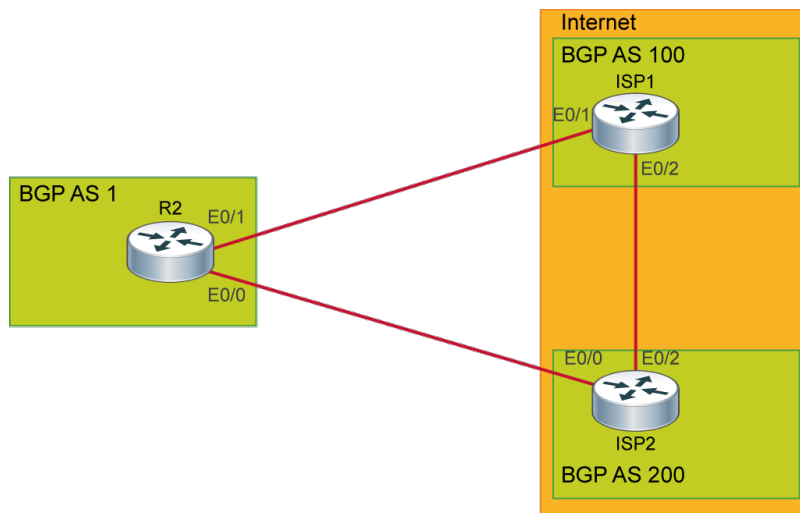
Discovery 11: Configure Per-Neighbor Weights

Overview

Through this discovery, you will learn how to configure per-neighbor weights and how to monitor [BGP](#) route selection. You will configure a multihomed customer's router, R2, to use the link to the primary service provider, ISP1, for all destinations.



Topology



Job Aids

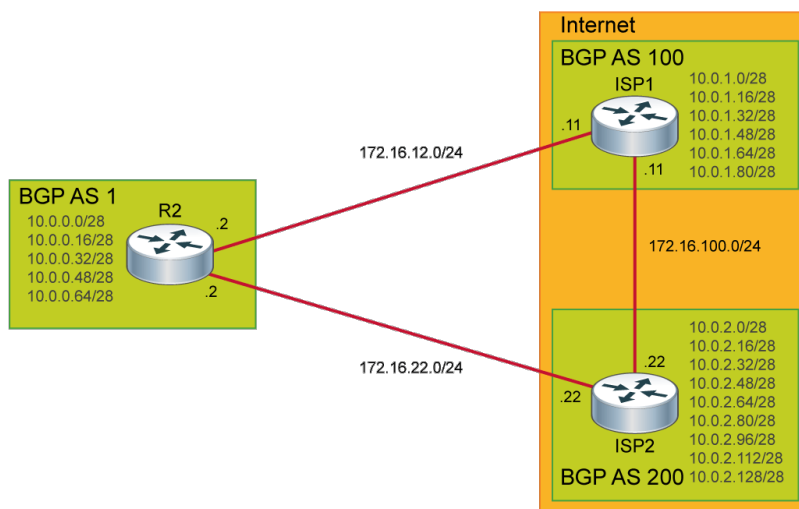


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Ethernet 0/2	172.16.100.11/24	Connection to ISP2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Ethernet 0/2	172.16.100.22/24	Connection to ISP1
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

IP addresses and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as EBGP:

- R2 to ISP1
- R2 to ISP2
- ISP1 to ISP2

Configure Per-Nighbor Weights

Discovery Steps

Step 1

On the R2 router, verify the initial state of default weights for all received destinations.

To verify weights, use the **show ip bgp** command.

```
R2# show ip bgp
BGP table version is 65, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28       0.0.0.0              0         32768 i
*> 10.0.0.16/28      0.0.0.0              0         32768 i
*> 10.0.0.32/28      0.0.0.0              0         32768 i
*> 10.0.0.48/28      0.0.0.0              0         32768 i
*> 10.0.0.64/28      0.0.0.0              0         32768 i
* 10.0.1.0/28        172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
* 10.0.1.16/28       172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
* 10.0.1.32/28       172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
* 10.0.1.48/28       172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
* 10.0.1.64/28       172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
   Network          Next Hop        Metric LocPrf Weight Path
* 10.0.1.80/28       172.16.22.22         0          0 200 100 i
*>                   172.16.12.11         0          0 100 i
* 10.0.2.0/28        172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.16/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.32/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.48/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.64/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.80/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.96/28       172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.112/28      172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
* 10.0.2.128/28      172.16.12.11         0          0 100 200 i
*>                   172.16.22.22         0          0 200 i
```

R2 has received routes to the following networks:

- 10.0.0.0/24 networks are injected into the BGP table by this router. By default, locally sourced routes are assigned a weight of 32768.
- 10.0.1.0/24 networks are received from ISP1 and ISP2. All routes have default weight value of 0 assigned.
- 10.0.2.0/24 networks are received from ISP1 and ISP2. All routes have default weight value of 0 assigned.

Step 2

On the R2 router, make sure that route received from the ISP1 router will be preferred over routes received from the ISP2 router.

For routes that are received from ISP1, configure a weight of 150 and for routes that are received from ISP2 configure a weight of 100.

Note that BGP routes with higher weight are preferred.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 weight 150
R2(config-router)# neighbor 172.16.22.22 weight 100
```

The weight is configured on R2 on both BGP sessions. A higher weight value is given to the routes that are received from the primary [ISP](#), ISP1, compared to the routes that are received from the backup ISP, ISP2.

Any time that the R2 receives routing information about the same IP network number from both ISPs, the router compares the weights that are assigned to the routes. The routes that are received from the ISP1 will always win this comparison.

Step 3

Re-establish BGP sessions on the R2 router to enforce new weights.

To re-establish BGP session, use the **clear ip bgp** command.

```
R2# clear ip bgp *
```

The * parameter of the command specifies to clear all peers.

Monitoring BGP Route Selection and Weights

```
router> show ip bgp
```

- Displays all BGP routes. Best routes marked with ">"
- With every displayed route, weight is associated.

```
router> show ip bgp ip-prefix [mask subnet-mask]
```

- Displays detailed information about all paths for a single prefix

Without any argument, the **show ip bgp** command displays the entire BGP table. Each route is displayed in one line. The network number is displayed, and if the subnet mask differs from the natural mask, the prefix length is indicated. The BGP next-hop attribute, [MED](#), local preference, weight, [AS](#) path, and origin code are displayed. Local preference is displayed only if it is not the default value.

The printout is sorted in network number order. If there is more than one route to the same network, the network number is printed on the first line only. The other routes to the same network have their network field left blank on the output. The greater-than (">") character indicates the routes that are selected as the best.

To get more detailed information about routes to a specific destination network, you can use the network number, and optionally the subnet mask, as an argument on the command line. These additions display more detailed information about that specific network.

First, a short summary that indicates the network number and prefix length is displayed, along with the table version number for this route. The next line indicates how many alternative routes have been received and which one of them the router has selected as the best. There are then a couple of lines for each of the received routes to reach the network. For each of the routes, all attributes are displayed. The one selected as the best also has the word "best" displayed

Step 4

On the R2 router, verify received routes to destination networks and their weights.

To verify received routes, use the **show ip bgp** command.

```
R2# show ip bgp
BGP table version is 20, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*>  10.0.0.0/28      0.0.0.0          0           32768 i
*>  10.0.0.16/28     0.0.0.0          0           32768 i
*>  10.0.0.32/28     0.0.0.0          0           32768 i
*>  10.0.0.48/28     0.0.0.0          0           32768 i
```


*>	10.0.0.64/28	0.0.0.0	0	32768	i
*	10.0.1.0/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
*	10.0.1.16/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
*	10.0.1.32/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
*	10.0.1.48/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
*	10.0.1.64/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
	Network	Next Hop	Metric	LocPrf	Weight Path
*	10.0.1.80/28	172.16.22.22		100	200 100 i
*>		172.16.12.11	0	150	100 i
*	10.0.2.0/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.16/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.32/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.48/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.64/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.80/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.96/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.112/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i
*	10.0.2.128/28	172.16.22.22		100	200 i
*>		172.16.12.11		150	100 200 i

R2 has received routes to 15 different networks outside of its own AS.

- 10.0.1.0/24 networks are received from ISP1 and ISP2.
- 10.0.2.0/24 networks are received from ISP1 and ISP2.

When the routes were received from the ISP1, the weight value 150 was assigned to each of the routes. When the routes were received from the ISP2, the weight value 100 was assigned to each of the routes. The R2 then makes the route selection—it has two alternative paths for each destination and it selects the path via the ISP1 as the best. It makes this selection based on the weight parameter, regardless of other BGP attributes, such as AS path length.

Consequently, the 10.0.2.0/24 networks that are directly connected to the ISP2 will also be reached via the ISP1. Although they are actually networks in the AS of the ISP2, AS 200.

Step 5

On the R2 router, verify detailed information about the 10.0.2.0/28 network.

To verify detailed information, use the **show ip bgp** command with a network number as an argument.

```
R2# show ip bgp 10.0.2.0/28
BGP routing table entry for 10.0.2.0/28, version 8
Paths: (2 available, best #2, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 2
  200
    172.16.22.22 from 172.16.22.22 (10.0.2.129)
      Origin IGP, metric 0, localpref 100, weight 100, valid, external
  Refresh Epoch 2
  100 200
    172.16.12.11 from 172.16.12.11 (10.0.1.81)
      Origin IGP, localpref 100, weight 150, valid, external, best
```

There are two alternatives to reach network 10.0.2.0/28. Each of them is received from different neighbors, one in AS 100 and another in AS 200. The network 10.0.2.0/28 is created in AS 200.

The route selection mechanism has selected the second route that is listed as the best. It was chosen because the weight value is higher.

Changing Weights with Route Maps

The route map is a powerful tool to select and alter routing information. When you apply a route map to incoming information from a BGP neighbor, each received update is examined as it passes through the route map. The sequence number of the statements specify the order of execution.

- Weights can be set with route maps in complex scenarios.
- Routes can be matched on any combination of prefix lists, AS path filters, or other BGP attributes.
- Routes not matched by the route-map are discarded.

```
router bgp as-number
  neighbor ip-address route-map route-map-name in
  !
route-map route-map-name permit sequence
  match condition
  set weight weight
```

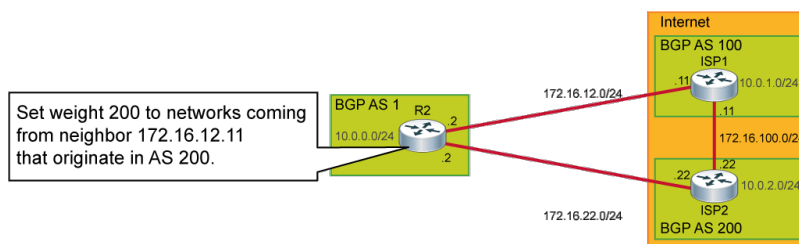
The first statement in the route map for which all the match clauses indicate a match is the one that is used. If the route map says "permit," the set clauses are applied to the route, the route is accepted, and the weight is changed.

Match clauses can be arbitrarily complex. One of them can refer to an AS path access list that does matching on AS paths. Another can refer to a prefix list that does matching on the announced network number. Only when all configured match clauses are evaluated, the route map statement is used and its result, permit, or deny, applied.

If a received route is not matched by any of the route map statements, the end of the route map is reached. The route map logic has an "implicit deny" rule, which means that if no statement selects a route, the route is discarded.

If you do not desire the "implicit deny" rule, add an "explicit permit all" at the end of the route map to overrule it. To ensure that such a route map statement is the last statement, you should assign it a very high sequence number. It should not have any match clause at all. The lack of a match clause means "match all." By not configuring any set clause, you can ensure that this statement does not alter any attributes.

Changing Weights with Route Maps Example



```
router bgp 1
  neighbor 172.16.12.11 route-map SetWeight200 in
  !
route-map SetWeight200 permit 10
  match as-path 47
  set weight 200
  !
route-map SetWeight200 permit 20
  set weight 100
  !
ip as-path access-list 47 permit _200$
```

All received routes from ISP1 neighbor have their AS paths checked against the AS path access list 47. AS path access list 47, as referenced by route map statement number 10, permits those routes with an AS path that indicates that they originated in AS 200. Routes that the route map statement number 10 in the SetWeight200 route map permits and selects, will have their weight set to 200. The set clause in the route map indicates the new weight value..

Route map statement number 20 then tests the routes that are not originated in AS

200 (routes that the AS path access list 47 does not permit). This statement does not include a match clause, indicating that all routes are matched. Therefore, the route map statement 20 match all routes that the route map statement does not. The route map has been configured with an “explicit permit all” statement at the end of the route map.

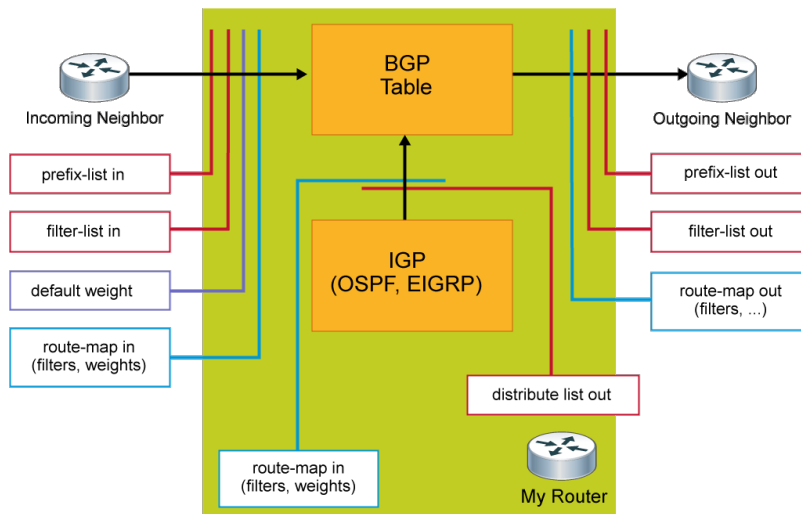
Routes that the route map statement 20 matches, have their weight set to 100. The result is that the router accepts the routes that originated in AS 200 (all 10.0.2.0/24 networks) and assigns the weight 200. All others are accepted and assigned the weight 100. This route map does not discard any route.



Specifying weights with filter lists is no longer supported in Cisco IOS Software Release 12.1 and later. The command has already been removed from Cisco IOS Software Release 12.1T. These releases use an incoming route map, where you match an AS path with the **match as-path** command and set the weight with the **set weight** command. When you are using a route map as a replacement for the filter list with the **weight** option, make sure that specifying a “**permit**” entry in the route map without an associated match condition does not filter all other routes.

BGP Route Selection and Filtering Tools Summary

Prefix lists and filter lists, both in and out, filter out routes and discard the ones that are not permitted. Weight setting is applicable only on incoming routes because a router never propagates the weight attribute to its neighbors. Route maps can be filters that discard routes but can also be used to modify and set various attributes on both incoming and outgoing routes.



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

Summary

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

- BGP uses a number of criteria for best-path route selection—the weight is taken into consideration first.
- The weight attribute can be assigned to a route using one of the following:
 - Default weight value can be assigned to all routes that are received from a specific neighbor.
 - Route maps can be applied to neighbors to set the weight attribute of received routes.
- You can use the **neighbor weight** command to assign a weight value to all routes that are received from a neighbor.
- You can use the **show ip bgp** command to display all BGP routes, the routes that BGP selected as best, and the weight attribute setting for each route.

Setting BGP Local Preference

Overview

Sometimes you need to influence [BGP](#) route selection. BGP provides many tools to achieve this. One of these tools is also the local preference attribute.

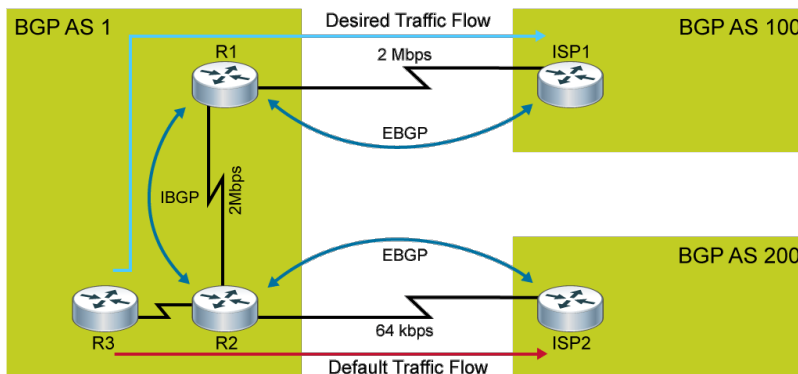
In this lesson, you will learn how to influence BGP route selection by setting the BGP local preference attribute of incoming BGP routes. Local preference is similar to the weight attribute but differs from the BGP weight attribute in that weight is local to a specific router on which it is configured. Two methods that are used to set the local preference attribute, default local preference and route maps, are discussed in this lesson. You will also learn how to monitor the BGP table to verify correct local preference configuration and to properly influence path selection.

Upon completing this lesson, you will be able to:

- Explain why using BGP weights might not provide consistent BGP route selection in an AS
- Describe how the BGP local preference attribute influences BGP route selection
- Identify the Cisco IOS command that is required to configure default BGP local preference on a router
- Identify the Cisco IOS commands that are required to monitor BGP local preference
- Describe how to configure default BGP local preference on a router
- Identify the Cisco IOS commands that are required to configure BGP local preference using route maps
- Describe how to configure BGP local preference using route-maps

Consistent Route Selection Within the AS

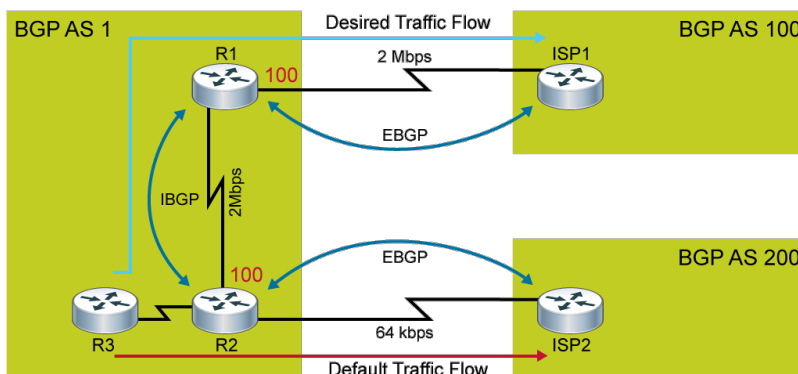
Using BGP in autonomous systems with a single neighbor relationship usually does not require any advanced features. In situations like the one shown in the figure, however, it is important to ensure that the customer routers choose the correct link. Obviously, the router should choose the 2-Mbps link and use the 64-kbps link only for backup purposes.



Q: Which routing protocol must be run in AS 1?

A: You must run IBGP in AS 1.

You have to make sure that the router selects the upper link (2-Mbps link) as its primary link and has the ability to switch over to the backup if a failure occurs. To assure this, you must configure an IBGP session between the two border routers in AS 1.

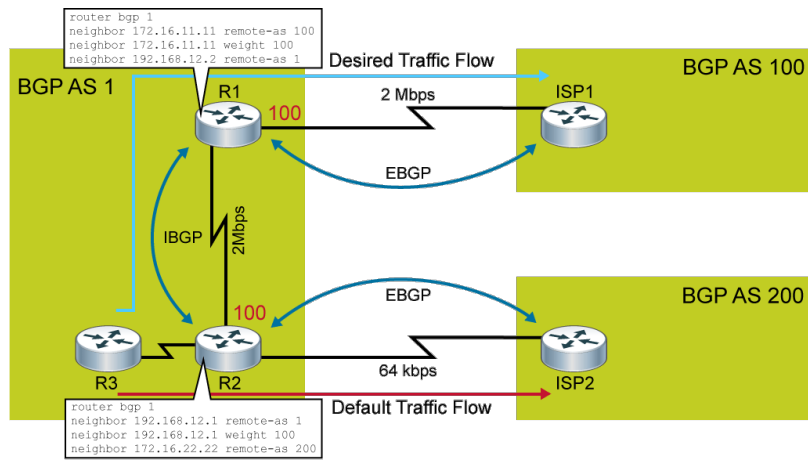


Q: How will you influence the route selection on routers in AS 1 so that they select the fastest route?

A: Use weights on EBGP and IBGP sessions.

One way of changing the default route selection is to use weights. Weight is an attribute that is locally significant to a router. Weight is a property, or parameter, and is therefore, not seen on any neighboring routers. When designing BGP networks using weights, you as a network administrator should set the weights on every router. If there is more than one path for the same network, a router will choose the one with the highest weight. The default value for weight is 0.

In this example, the R1 router in AS 1 sets a weight of 100 for routes that are received over the 2-Mbps link from AS 100 (primary link). R1 prefers these routes to possible internal updates from the bottom router, where the default weight is 0. The bottom router sets a weight of 100 for internal routes that are received from the upper router and prefers them to routes that are received from AS 200. As a result, all packets will leave the AS through the primary 2-Mbps link.



Run the traffic over the fastest line available.

The configurations that are shown in the figure demonstrate how to change the default weight on a per-neighbor basis. If you use the **neighbor weight** command, all newly arrived updates will have a weight of 100. Updates coming from the other neighbor will still have the default weight of 0.

After you have applied the **neighbor weight** command, a refresh is needed from the neighbor. There are three ways of doing this, depending on the Cisco IOS version:

- Use the **clear ip bgp neighbor address** command to clear the neighbor relationship and re-establish it to refresh the BGP entries and apply the weight.
- Configure soft reconfiguration for the neighbor and use the **clear** command. You can perform all subsequent clearing by using the **clear ip bgp neighbor address soft in** command, which does not reset the neighbor relationship. The Cisco IOS Software Release 11.2 and above supports the soft reconfiguration feature..
- Use the **clear ip bgp neighbor address in** command if both neighboring routers support the route refresh. The Cisco IOS Software Release 12.1 and above supports the route refresh feature.

When you are trying to implement this example with weights, it requires two route maps on each router within AS 1. Luckily, BGP has a similar mechanism that you can use for consistent AS-wide route selection: local preference.

BGP Local Preference

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.

- You can use local preference to ensure AS-wide route selection policy.
- Any BGP router can set local preference when processing incoming route updates, when doing redistribution, or when sending outgoing route updates.
- Local preference is used to select routes with equal weight.
- Local preference is stripped in outgoing EBGp updates except in EBGp updates with confederation peers.

Local preference is the second-strongest criterion in the route selection process. If there are two or more paths available for the same network, a router will first compare weight. 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.

- Local preference is the second strongest BGP route selection parameter.
- 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 that are 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.

Local preference is the second-strongest [BGP](#) route selection parameter.

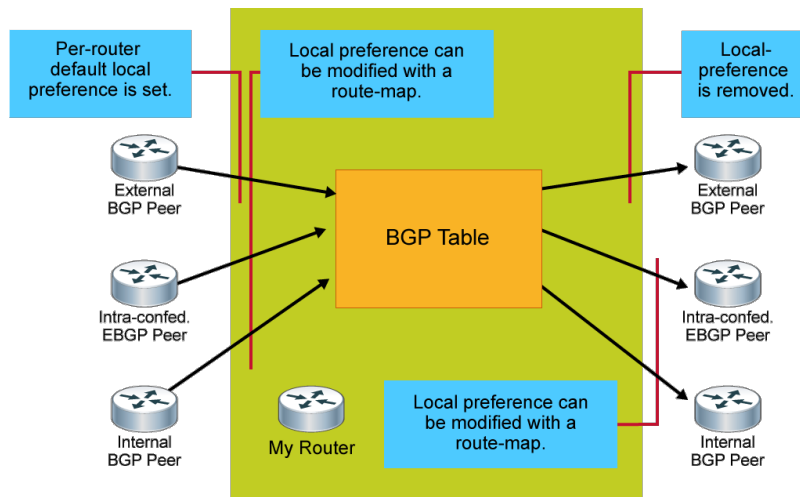
Remember the route selection rules:

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 you as an administrator can use both weight and local preference to manipulate the route selection process, you must decide which one to use. If local preference is used, the weight should be the same for all paths.

You 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. You should avoid changing the local preference attribute on internal sessions to prevent unnecessary complexity and unpredictable behavior.



You can apply local preference in the following ways:

- Use a route map with the **set local-preference** command. You can use the route map on incoming updates from all neighbors or on outgoing updates to internal neighbors (not recommended).
- Use the **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.

Configuring Default Local Preference

You can use the **bgp default local-preference** command in [BGP](#) configuration mode to change the default value of local preference. The new default value applies only to locally originated routes and the routes that are received from external neighbors.

```
router(config-router)# bgp default local-preference preference
```

- This command changes the default local preference value.
- The specified value is applied to all routes that do not have local preference set (EBGP routes).
- The default value of this parameter is 100, allowing you to specify more desirable or less desirable routers.

If you set a value lower than the default of 100, the router will prefer internal paths to external (normally a router would prefer external routes).

If you set a value higher than 100, then external paths will be preferred to all internal paths (also the ones with a shorter [AS](#) path).

Monitoring Local Preference

- Display nondefault local preference with the **show ip bgp** command.
- Local preference is displayed in the **show ip bgp prefix** command.
- Local preference is also displayed in [BGP](#) update debugging (only for inbound updates).

Although local preference is not a mandatory attribute, it is applied to every route. When you are using the **show ip bgp** command, a locally applied default value is not shown. All other values are displayed. You should use the **show ip bgp prefix** command to also display the locally applied value.



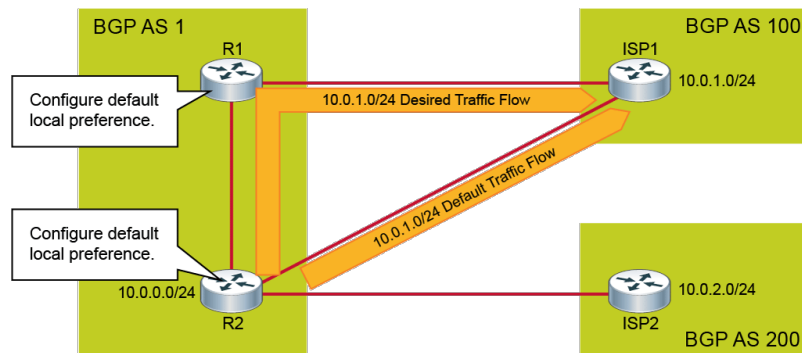
The output of the **show ip bgp** command will not display the local preference value if the value is the same as the **bgp default local-preference value** in the local router.

The output that is displayed from **show** and **debug** commands will vary depending on the Cisco IOS version. Newer versions typically display more information. In Cisco IOS Software Release 12.0 and in later versions, enabling debugging of incoming routing updates will also display the local preference attribute.

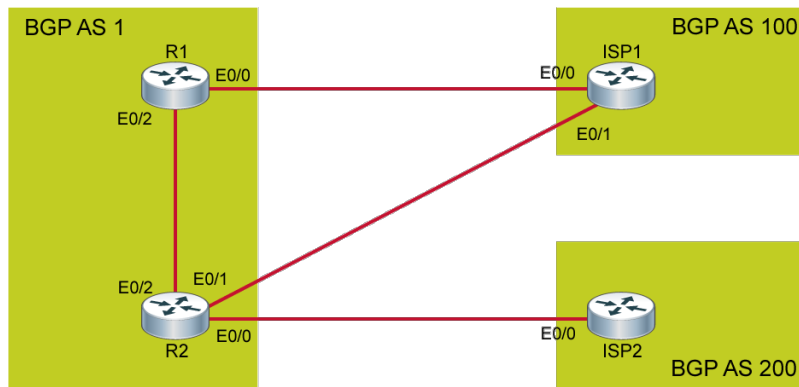
Discovery 12: Configure and Monitor Local Preference

Overview

Through this discovery, you will learn how to configure [BGP](#) default local preference. You will influence R2 router to prefer the path to ISP1 networks via R1, instead of using the directly connected link.



Topology



Job Aids

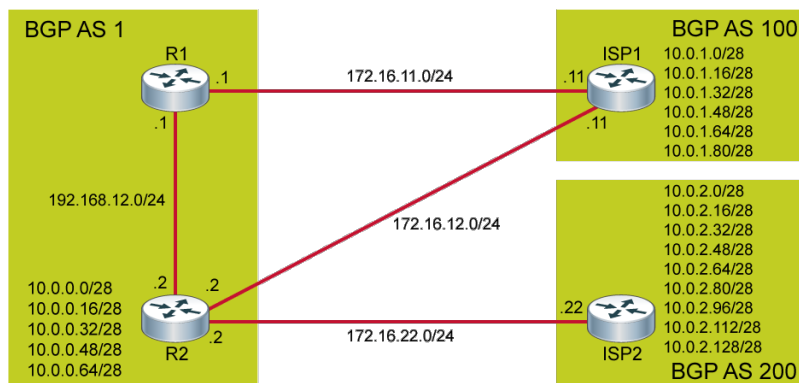


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/0	172.16.11.11/24	Connection to R1
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured:

- [EBGP](#)
 - R1 to ISP1
 - R2 to ISP1
 - R2 to ISP2

- [IBGP](#)
 - R1 to R2

Configure and Monitor Local Preference

Discovery Steps

Step 1

On the R2 router, verify the initial state of default local preferences for all received destinations.

```
R2# show ip bgp
BGP table version is 21, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	10.0.0.0/28	0.0.0.0	0		32768	i
*>	10.0.0.16/28	0.0.0.0	0		32768	i
*>	10.0.0.32/28	0.0.0.0	0		32768	i
*>	10.0.0.48/28	0.0.0.0	0		32768	i
*>	10.0.0.64/28	0.0.0.0	0		32768	i
* i	10.0.1.0/28	192.168.12.1	0	100	0	100 i
*>		172.16.12.11	0		0	100 i
* i	10.0.1.16/28	192.168.12.1	0	100	0	100 i
*>		172.16.12.11	0		0	100 i
* i	10.0.1.32/28	192.168.12.1	0	100	0	100 i
*>		172.16.12.11	0		0	100 i
* i	10.0.1.48/28	192.168.12.1	0	100	0	100 i
*>		172.16.12.11	0		0	100 i
* i	10.0.1.64/28	192.168.12.1	0	100	0	100 i
	Network	Next Hop	Metric	LocPrf	Weight	Path
*>		172.16.12.11	0		0	100 i
* i	10.0.1.80/28	192.168.12.1	0	100	0	100 i
*>		172.16.12.11	0		0	100 i
*>	10.0.2.0/28	172.16.22.22	0		0	200 i
*>	10.0.2.16/28	172.16.22.22	0		0	200 i
*>	10.0.2.32/28	172.16.22.22	0		0	200 i
*>	10.0.2.48/28	172.16.22.22	0		0	200 i
*>	10.0.2.64/28	172.16.22.22	0		0	200 i
*>	10.0.2.80/28	172.16.22.22	0		0	200 i
*>	10.0.2.96/28	172.16.22.22	0		0	200 i
*>	10.0.2.112/28	172.16.22.22	0		0	200 i
*>	10.0.2.128/28	172.16.22.22	0		0	200 i

R2 has received routes to the following networks:

- 10.0.0.0/24 networks (originating locally on R2), for which the applied default local preference of 100 is not displayed.
- 10.0.1.0/24 networks (originating on the ISP1 router) are received from R1 and ISP1:
 - The first path for these networks was received from R1 (with next hop 192.168.12.1), with local preference value of 100. Local preference of internal route is displayed in the output.
 - The second path for these networks was received from ISP1 (with next hop 172.16.12.11). The applied default local preference of 100 is not displayed.
- 10.0.2.0/24 networks (originating on the ISP2 router) are received from ISP2. The applied default local preference of 100 is not displayed.

For 10.0.1.0/24 networks, multiple paths exist and the decision mechanism on R2 prefers the external (EBGP) paths over internal (IBGP).

Step 2

Influence the R2 router to prefer the path to ISP1 networks via R1 router.

On R1, apply local preference of 120 for all routes that are received from external neighbor.

```
R1(config)# router bgp 1
R1(config-router)# bgp default local-preference 120
```

Step 3

On the R2 router, verify configured default local preference for all received destinations.

```

R2# show ip bgp
BGP table version is 20, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop           Metric LocPrf Weight Path
*> 10.0.0.0/28     0.0.0.0               0         32768 i
*> 10.0.0.16/28    0.0.0.0               0         32768 i
*> 10.0.0.32/28    0.0.0.0               0         32768 i
*> 10.0.0.48/28    0.0.0.0               0         32768 i
*> 10.0.0.64/28    0.0.0.0               0         32768 i
*>i 10.0.1.0/28     192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
*>i 10.0.1.16/28    192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
*>i 10.0.1.32/28    192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
*>i 10.0.1.48/28    192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
*>i 10.0.1.64/28    192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
   Network        Next Hop           Metric LocPrf Weight Path
*>i 10.0.1.80/28    192.168.12.1          0        120         0 100 i
*                  172.16.12.11          0         0 100 i
*> 10.0.2.0/28     172.16.22.22          0         0 200 i
*> 10.0.2.16/28    172.16.22.22          0         0 200 i
*> 10.0.2.32/28    172.16.22.22          0         0 200 i
*> 10.0.2.48/28    172.16.22.22          0         0 200 i
*> 10.0.2.64/28    172.16.22.22          0         0 200 i
*> 10.0.2.80/28    172.16.22.22          0         0 200 i
*> 10.0.2.96/28    172.16.22.22          0         0 200 i
*> 10.0.2.112/28   172.16.22.22          0         0 200 i
*> 10.0.2.128/28   172.16.22.22          0         0 200 i

```

R2 has received routes to the following networks:

- 10.0.0.0/24 networks that originate locally on R2, and the applied default local preference of 100 is not displayed.
- 10.0.1.0/24 networks are received from R1 and ISP1:
 - The first path for these networks was received from R1 (next hop 192.168.12.1), with a changed local preference value of 120. Local preference of internal route is displayed in the output. Note that this is the best path for 10.0.1.0/24 networks, marked with > sign.
 - The second path for these networks was received from ISP1 (next hop 172.16.12.11). The applied default local preference of 100 is not displayed.
- 10.0.2.0/24 networks are received from ISP2. The applied default local preference of 100 is not displayed.

For 10.0.1.0/24 networks, multiple paths exist and the decision mechanism on R2 selects a route with higher local preference attribute (120), since weight value is left on default, 0.

Configuring Local Preference with Route Maps

To have more control over setting local preference, you may be forced to use a route map.

```
router(config)#  
route-map name permit sequence  
  match condition  
  set local-preference value
```

- Changes BGP local preference only for routes that the route map entry matches

```
router(config-router)# neighbor address route-map name in | out
```

- Applies route map to incoming updates from specified neighbor or outgoing updates to specified neighbor
- Per-neighbor local preference that is configured by using a route map with no match condition

A route map can have more sequenced statements, each with a different **set local-preference** command, and a different **match** condition. If there is no **match** command, the route map statement will apply local preference to all routes. You can then apply the route map to [BGP](#) route updates in either the incoming or outgoing direction.



Applying a route map to outgoing updates on external sessions will have no effect on local preference in the neighboring AS.

When you use a route map to set local preference, the route map is typically applied to incoming BGP routes that the [EBGP](#) neighbor advertises. The local router uses the local preference attribute in BGP route selection. In addition, the router also propagates the attribute to all [IBGP](#) sessions in the local [AS](#). Normally, no modifications of local preference are made on IBGP sessions. This restriction ensures that all routers in the local AS use the same local preference value and make the same decision in the route selection process.

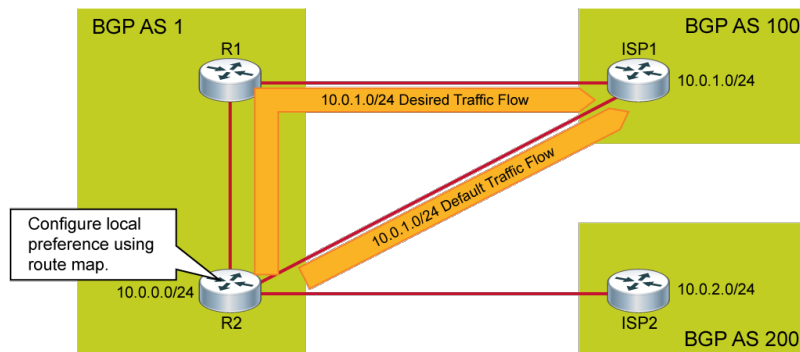


If a network is not matched in any of the route map statements, the network will be filtered. To permit unmatched networks without setting the local preference attribute, you should add another route map statement without **match** or **set** commands to the end of the route map. This statement should simply permit the remaining networks.

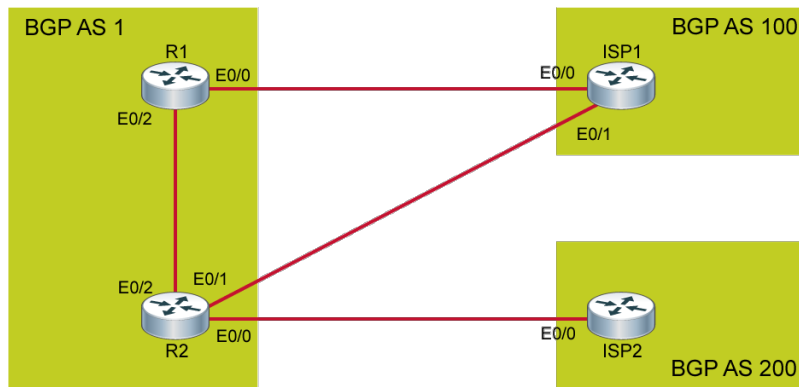
Discovery 13: Configure Local Preference Using Route Maps

Overview

Through this discovery, you will learn how to configure local preference using route maps. You will again configure R2 router to prefer the path to ISP1 networks via R1, instead of using the directly connected link.



Topology



Job Aids

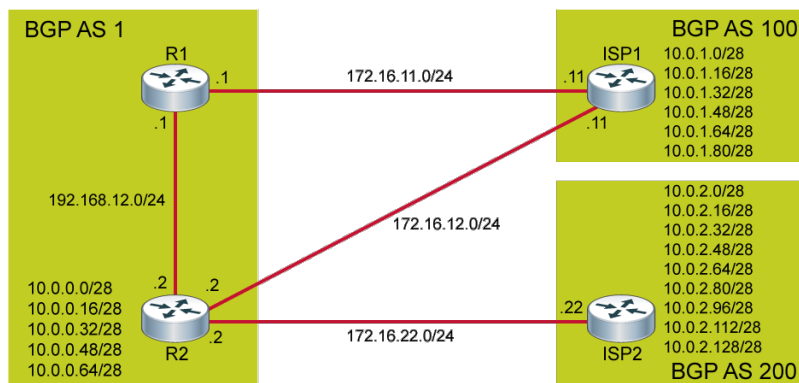


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/0	172.16.11.11/24	Connection to R1
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured:

- [EBGP](#)
 - R1 to ISP1
 - R2 to ISP1
 - R2 to ISP2

- [IBGP](#)
 - R1 to R2

Configure Local Preference Using Route Maps

Discovery Steps

Step 1

On the R2 router, verify the initial state of default local preferences for all received destinations.

To verify local preferences, use the **show ip bgp** command.

```
R2# show ip bgp
BGP table version is 21, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28       0.0.0.0            0         32768 i
*> 10.0.0.16/28      0.0.0.0            0         32768 i
*> 10.0.0.32/28      0.0.0.0            0         32768 i
*> 10.0.0.48/28      0.0.0.0            0         32768 i
*> 10.0.0.64/28      0.0.0.0            0         32768 i
* i 10.0.1.0/28      192.168.12.1       0        100      0 100 i
*>                172.16.12.11       0         0 100 i
* i 10.0.1.16/28     192.168.12.1       0        100      0 100 i
*>                172.16.12.11       0         0 100 i
* i 10.0.1.32/28     192.168.12.1       0        100      0 100 i
*>                172.16.12.11       0         0 100 i
* i 10.0.1.48/28     192.168.12.1       0        100      0 100 i
*>                172.16.12.11       0         0 100 i
* i 10.0.1.64/28     192.168.12.1       0        100      0 100 i
   Network          Next Hop        Metric LocPrf Weight Path
*>                172.16.12.11       0         0 100 i
* i 10.0.1.80/28     192.168.12.1       0        100      0 100 i
*>                172.16.12.11       0         0 100 i
*> 10.0.2.0/28       172.16.22.22       0         0 200 i
*> 10.0.2.16/28      172.16.22.22       0         0 200 i
*> 10.0.2.32/28      172.16.22.22       0         0 200 i
*> 10.0.2.48/28      172.16.22.22       0         0 200 i
*> 10.0.2.64/28      172.16.22.22       0         0 200 i
*> 10.0.2.80/28      172.16.22.22       0         0 200 i
*> 10.0.2.96/28      172.16.22.22       0         0 200 i
*> 10.0.2.112/28     172.16.22.22       0         0 200 i
*> 10.0.2.128/28     172.16.22.22       0         0 200 i
```

R2 has received routes to the following networks:

- 10.0.0.0/24 networks that originate locally on R2, and the applied default local preference of 100 is not displayed.
- 10.0.1.0/24 networks are received from R1 and ISP1:
 - The first path for these networks was received from R1 (next hop 192.168.12.1), with local preference value of 100. Local preference of internal route is displayed in the output.
 - The second path for these networks was received from ISP1 (next hop 172.16.12.11). The applied default local preference of 100 is not displayed.
- 10.0.2.0/24 networks are received from ISP2. The applied default local preference of 100 is not displayed.

For 10.0.1.0/24 networks, multiple paths exist and the decision mechanism on R2 prefers the external (EBGP) paths over internal (IBGP).

Step 2

On the R2 router, configure route map to set local preference on 10.

Name the route map "LP10."

```
R2(config)# route-map LP10 10
R2(config-route-map)# set local-preference 10
```

Step 3

Influence the R2 router to prefer the path to ISP1 networks via R1 router. On the R2 router, use configured route map to set lower local preference for routes that

are received from AS 100.

Apply route map LP10 for routes that are received from AS 100.

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 route-map LP10 in
```

Local preference of 10 will now be applied to external updates from ISP1. Consequently, R2 will prefer the path to networks in AS 100 via R1, since it has higher local preference value.

Step 4

Clear all BGP sessions on the R2 router.

```
R2# clear ip bgp *
```

Step 5

On the R2 router, verify local preference for all received destinations.

```
R2# show ip bgp
BGP table version is 20, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop      Metric LocPrf Weight Path
*> 10.0.0.0/28       0.0.0.0          0         32768 i
*> 10.0.0.16/28      0.0.0.0          0         32768 i
*> 10.0.0.32/28      0.0.0.0          0         32768 i
*> 10.0.0.48/28      0.0.0.0          0         32768 i
*> 10.0.0.64/28      0.0.0.0          0         32768 i
*>i 10.0.1.0/28      192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
*>i 10.0.1.16/28     192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
*>i 10.0.1.32/28     192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
*>i 10.0.1.48/28     192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
*>i 10.0.1.64/28     192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
   Network          Next Hop      Metric LocPrf Weight Path
*>i 10.0.1.80/28     192.168.12.1     0        100      0 100 i
*                   172.16.12.11     0         10      0 100 i
*> 10.0.2.0/28       172.16.22.22     0         0      200 i
*> 10.0.2.16/28      172.16.22.22     0         0      200 i
*> 10.0.2.32/28      172.16.22.22     0         0      200 i
*> 10.0.2.48/28      172.16.22.22     0         0      200 i
*> 10.0.2.64/28      172.16.22.22     0         0      200 i
*> 10.0.2.80/28      172.16.22.22     0         0      200 i
*> 10.0.2.96/28      172.16.22.22     0         0      200 i
*> 10.0.2.112/28     172.16.22.22     0         0      200 i
*> 10.0.2.128/28     172.16.22.22     0         0      200 i
```

R2 has received routes to the following networks:

- 10.0.0.0/24 networks that originate locally on R2, and the applied default local preference of 100 is not displayed.
- 10.0.1.0/24 networks are received from R1 and ISP1:
 - The first path for these networks was received from R1 (next hop 192.168.12.1) with a default local preference value of 100. Local preference of internal route is displayed in the output.
 - The second path for these networks was received from ISP1 (next hop 172.16.12.11) with a local preference value of 10, set by the route map. The applied local preference is not left on default and is displayed in the output.
- 10.0.2.0/24 networks are received from ISP2. The applied default local preference of 100 is not displayed.

For 10.0.1.0/24 networks, multiple paths exist and the decision mechanism on R2 selects

the route with a higher local preference attribute (100), since weight value is left on default, 0. Consequently the path via R1 will be chosen to reach 10.0.1.0/24 networks.

Summary

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

- Local preference is similar to the weight attribute. You can use both to influence BGP path selection, but the weight is local to the specific router on which it is configured.
- You can use local preference to ensure AS-wide route selection policy because it can be seen on neighboring routers without the need to reset it.
- You should avoid mixing weight and local preference because weight has priority when you are selecting the best path.
- Configure local preference:
 - Using the **bgp default local-preference** *preference* command.
 - Using route map statements.

Using AS Path Prepending

Overview

[BGP](#) provides many tools for administrators to influence route selection. You have already learned how to influence BGP route selection for outgoing traffic by setting weight or local preference.

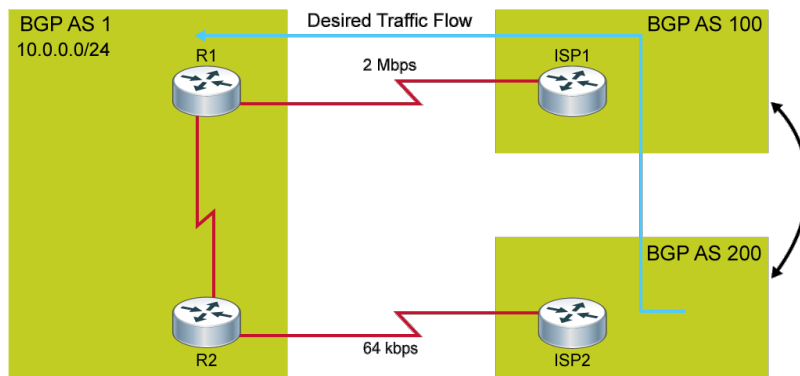
Problems can arise when administrative policies mandate a specific return path to be used for traffic returning to the [AS](#). AS path prepending potentially allows the customer to influence the route selection of its service providers, so the path for returning traffic. In this lesson, you will learn about AS path prepending and the Cisco IOS commands that are required to properly configure and monitor AS path configurations. You will also learn filtering requirements for influencing route selection using AS path prepending.

Upon completing this lesson, you will be able to:

- Explain the need to influence BGP return path selection in a service provider environment
- Describe the function of AS path prepending and how you can use it to facilitate proper return path selection
- Identify design considerations when you are implementing AS path prepending to influence return path selection
- Describe how to configure AS path prepending on the customer side and AS path filtering on the service provider side
- Describes how the BGP Hide Local-Autonomous System feature simplifies the task of changing the AS number in a BGP network

Return Path Selection in a Multihomed AS

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

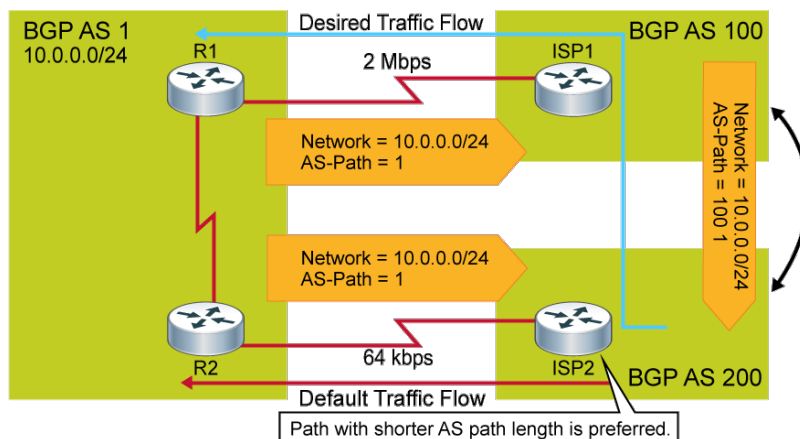


- **Requirement:** The return traffic to the customer must arrive over the highest-speed access link.

If you configure the preferred path only for outgoing traffic and not for incoming (return) traffic, you will likely get an asymmetrical traffic flow. It can as well result in 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 200 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 that the network could achieve.

In this example, AS 1 requests AS 200 to send packets toward network 10.0.0.0/24 via AS 100. The reason for this request is to improve the network performance and minimize delay. Assuming, of course, that the connectivity between AS 100 and AS 200 is better than the direct 64-kbps link between AS 200 and AS 1.

Default Return Path



- **Result:** The return traffic flows over the path with the shortest AS path length.

If you do not configure any BGP path selection tools on the route to influence the traffic flow, AS 200 will use the shortest AS path. This action will result in unwanted behavior because the return traffic to AS 1 will be sent over the low-speed WAN link.

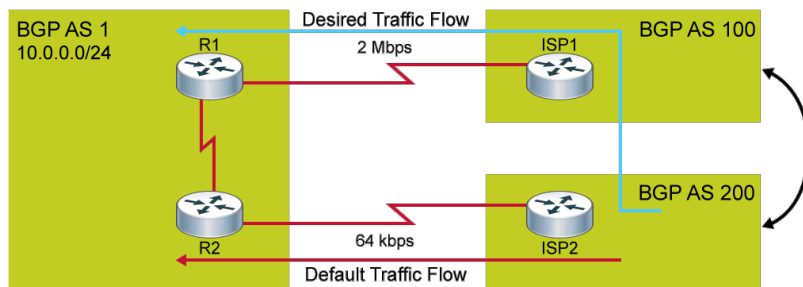
AS 1 announces network 10.0.0.0/24 over EBGp sessions to both AS 100 and AS 200. When AS 1 sends EBGp updates, it changes the AS path attribute according to BGP specifications. Both AS 100 and AS 200 receive a BGP update for network 10.0.0.0/24 with the AS path set to 1.

AS 100 selects the route for network 10.0.0.0/24 that it received from AS 1 as its best route. AS 100 uses that route and forwards it to AS 200. According to BGP specifications, AS 100 also changes the AS path attribute. AS 200 receives the route to network 10.0.0.0/24 from AS 100 with an AS path set to 100 1.

AS 200 has now received two alternative routes to network 10.0.0.0/24 (the direct

route from AS 1 and the route through AS 100). Remember that you did not configure anything in AS 200 to influence the flow of traffic. So, the router will use the BGP route selection rule of shortest AS path to select the best return path to network 10.0.0.0/24.

Proper Return Path Selection



Q: How do you select the proper return path from AS 200?

A: Use local preference in AS 200.

Q: Will the administrator of AS 200 configure it?

A: Unlikely.

Remember that the incoming traffic flow (from the perspective of AS 1) will be a result of the route selection for outgoing traffic in AS 200. The traffic that is going out from AS 200 will end up as incoming traffic in AS 1.

AS 200 can configure some changes that cause the route selection process for outgoing traffic to prefer to reach network 10.0.0.0/24 via AS 100. These changes would result in behavior matching the desired administrative policy for AS 1, 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 1 is to reconfigure the router in AS 200, which is receiving EBGp updates directly from AS 1. You can configure the router in AS 200 to assign a local preference value less than the default value (100) to all routes that are received from AS 1. The router in AS 200 is also configured specifically not to set local preference on EBGp routes that are received from AS 100. This configuration results in assignment of the default value of 100 to all routes received from AS 100. This difference in local preference values causes AS 200 routers to select the path via AS 100 as the best one to reach 10.0.0.0/24 networks.

However, all the configuration work to complete this process must be performed in AS 200. The network administrators of AS 200 would be required to modify the router configurations in AS 200 to satisfy the administrative policy requirements of AS 1. All changes must be documented and maintained according to the rules and procedures that have been adopted by AS 200.

If AS 200 is a major [ISP](#), the network administrators most likely are 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.

BGP Route Selection Rules

- BGP route selection uses the following 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 could result in proper return path selection.

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 route selection criteria.

It is unlikely that you, as the operator of an AS, 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. 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 originate at the router that is performing the route selection. The fourth criterion will apply, however, because the AS paths have different lengths.

If you do not manually manipulate the AS path by some administrative means, the router selects the path going over the fewest number of ASs, regardless of the 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.

AS Path Prepending

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 undesired return path. The AS path sent out over the undesired link becomes longer than the AS path sent out over the preferred path. So, the undesired link is now less likely to be used as the return path.

AS path prepending is known as manual manipulation of AS path length.

- The AS path should be extended with multiple copies of the AS number of the sender.
- AS path prepending is used to:
 - Ensure proper return path selection.
 - Distribute the return traffic load for multihomed customers.

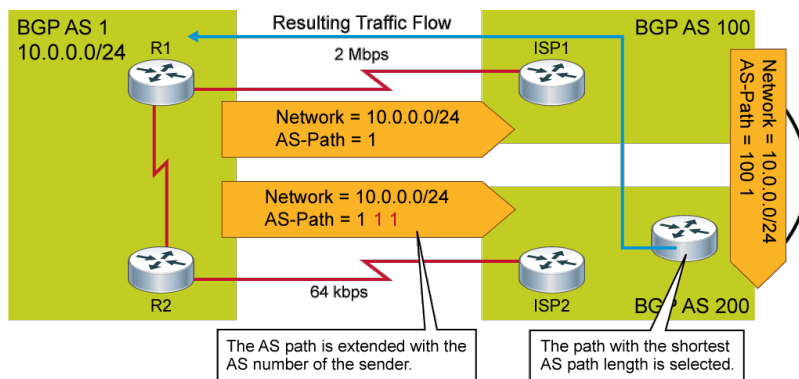
The length of the AS path is extended because extra copies of the AS number of the sender are prepended to (added to the beginning of) the AS path attribute. To avoid clashes with BGP loop prevention mechanisms, you should prepend no other AS number to the AS path attribute, except that of the sending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

If you prepend another AS number 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 Path Prepending Example

Administrative policy in AS 1 prefers that the low-speed link is used for backup purposes only. As long as the high-speed link between AS 1 and AS 100 is available, all traffic should flow toward AS 1 using the high-speed link.



- **Result:** The return traffic flows over the desired return path.

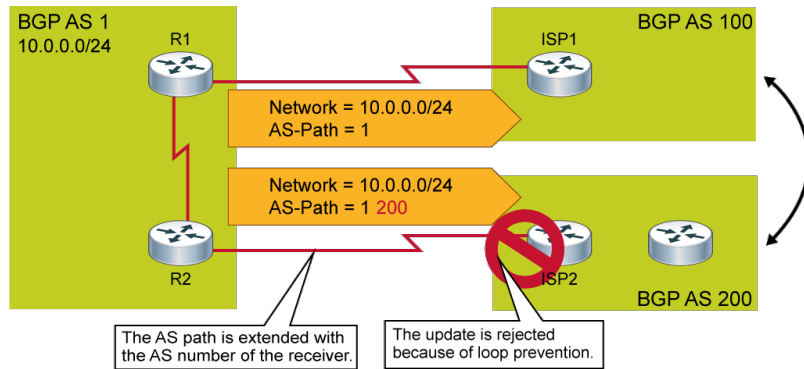
To accomplish this goal, you can configure the router in AS 1, that sends EBGp updates to AS 200, to prepend two copies of AS number "1" to AS path.

AS 200 receives two alternative routes to reach network 10.0.0.0/24:

- The update that it has received directly from AS 1 and has a manipulated AS path with a length of three.
- The update that it has received via AS 100 and was not manually manipulated, and therefore contains an AS path length of two.

When AS 200 starts the selection process for best route to reach network 10.0.0.0/24, it checks the AS path length after the weight and local preference parameters. In this case, neither weight nor local preference has been configured, the length of the AS path will be the deciding factor in the route selection process. Consequently, AS 200 prefers the shortest AS path and thus forwards packets toward network 10.0.0.0/24 via AS 100. The desired administrative policy has been met, and AS 1 will receive incoming traffic over the high-speed link.

If the forwarding path from AS 200 via AS 100 to AS 1 and network 10.0.0.0/24 is later broken, the BGP update to reach network 10.0.0.0/24 is revoked. In case of such a network failure, AS 200 will have only one remaining path to reach network 10.0.0.0/24. The route selection process now has only one choice, the route directly to AS 1 over the low-speed WAN link. The low-speed link will therefore serve as backup to the high-speed WAN link.



- Prepend the AS path with the AS number of the sender, not the AS number of the receiver.

When you are manually manipulating AS paths, the only valid AS number that you can prepend is the AS number of the sender. Prepending any other AS number will cause problems.

In the example, AS 1 is prepending AS number 200. The egress router performs AS path prepending when the route is on its way to be transmitted to AS 200. After the manual manipulation, BGP automatically changes the AS path according to the BGP specifications. The local AS number should always be added first when updates are sent over an EBGP session. Therefore, when AS 200 receives the BGP update, the AS path contains the value 1 200. The AS number 200 was set by the manual manipulation, and the AS number 1 was prepended automatically by BGP because the update was sent over an EBGP session.

When the edge router in AS 200 receives the BGP update, it checks the AS path to verify that the BGP updates were not propagated accidentally by a routing loop. Because the edge router finds its own AS number in the AS path, it assumes that the BGP update has already been in AS 200. According to the BGP specification, the update will be silently ignored.

Now assume that AS 1 had, for the manual manipulation, used a different AS number, not its own and not AS number 200. Would AS 200 now have accepted the update? The answer is yes. However, in this scenario, a problem would have appeared at a later stage when the route finally reached the actual AS belonging to the manually prepended AS number. This AS would have rejected the route because it would have found its own AS number somewhere in the AS path.

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.

There is no exact mechanism to calculate the required prepended AS path length.

- If a primary and backup scenario is desired:
 - 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 improved security (unexpected changes in the Internet).
- If traffic load distribution is desired:
 - Start with a short prepended AS path, monitor link use, and extend the prepended path length as needed.
 - Continuously monitor the link use and change the prepended AS path length if required.

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

- **Establishing a primary/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, routers forward only routes that are selected as best. So an AS that receives multiple alternatives to a destination will select the route with the shortest AS path and forward only that route.

When 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 then advertised through the Internet. In the case where 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 announced AS path to the [EBGP](#) neighbor connected via the backup link, the less likely it is that incoming traffic will be received from that neighbor. You, as a network administrator, can make a clever guess about how many copies of the AS number to prepend. After the prepending is implemented, you have to examine the result. If the expected result is not achieved, you can change the configuration and prepend a few more copies of the AS number.

After AS path prepending has generated the desired results, you 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 exactly predetermine the 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 (which autonomous systems they belong to). 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.

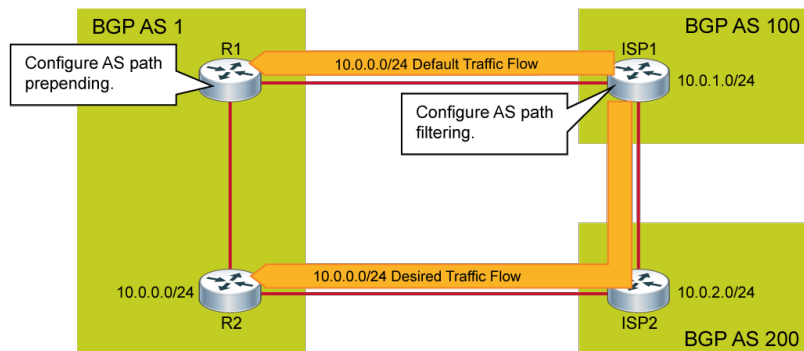
In a first attempt, you can configure a router that is connected to an overused link to prepend only a few extra copies of the local AS number. After the network has been given time to converge, you must check the change in load distribution. Monitoring of the load must be done for a period long enough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

copies of the local AS number. Then the process of resending local routes and monitoring the results starts all over again.

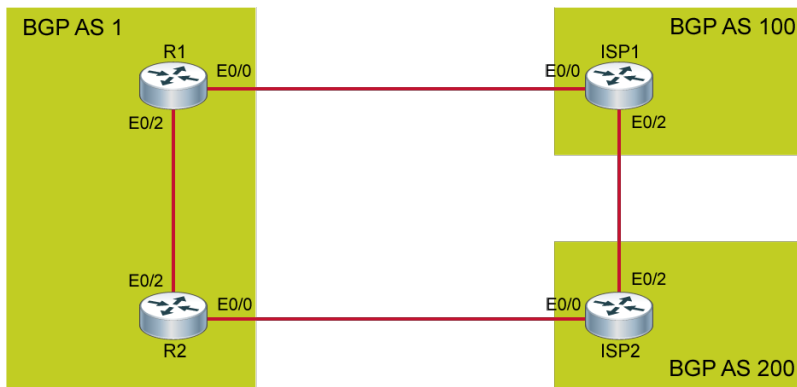
Discovery 14: Configure AS Path Prepending

Overview

In this discovery, you will first configure [AS](#) path prepending on the customer side. Then you will for a moment imagine that you are on a service provider side and you will configure AS path filtering on the service provider side.



Topology



Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/0	172.16.11.11/24	Connection to R1
ISP1	Ethernet 0/2	172.16.100.11/24	Connection to ISP2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Ethernet 0/2	172.16.100.22/24	Connection to ISP1
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured:

- [EBGP](#)
 - R1 to ISP1
 - R2 to ISP2
 - ISP1 to ISP2

- [IBGP](#)
 - R1 to R2

Configure AS Path Prepending

Discovery Steps

Step 1

On the ISP1 router, verify initial best path to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 20, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
* 10.0.0.0/28     172.16.100.22      0      0 200 1 i
*> 10.0.0.0/28     172.16.11.1        0      0 1 i
* 10.0.0.16/28    172.16.100.22      0      0 200 1 i
*> 10.0.0.16/28    172.16.11.1        0      0 1 i
* 10.0.0.32/28    172.16.100.22      0      0 200 1 i
*> 10.0.0.32/28    172.16.11.1        0      0 1 i
* 10.0.0.48/28    172.16.100.22      0      0 200 1 i
*> 10.0.0.48/28    172.16.11.1        0      0 1 i
* 10.0.0.64/28    172.16.100.22      0      0 200 1 i
*> 10.0.0.64/28    172.16.11.1        0      0 1 i
*> 10.0.1.0/28     0.0.0.0            0      32768 i
*> 10.0.1.16/28    0.0.0.0            0      32768 i
*> 10.0.1.32/28    0.0.0.0            0      32768 i
*> 10.0.1.48/28    0.0.0.0            0      32768 i
*> 10.0.1.64/28    0.0.0.0            0      32768 i
*> 10.0.1.80/28    0.0.0.0            0      32768 i
   Network        Next Hop        Metric LocPrf Weight Path
* 10.0.2.0/28     172.16.11.1        0      0 1 200 i
*> 10.0.2.0/28     172.16.100.22      0      0 200 i
* 10.0.2.16/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.16/28    172.16.100.22      0      0 200 i
* 10.0.2.32/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.32/28    172.16.100.22      0      0 200 i
* 10.0.2.48/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.48/28    172.16.100.22      0      0 200 i
* 10.0.2.64/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.64/28    172.16.100.22      0      0 200 i
* 10.0.2.80/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.80/28    172.16.100.22      0      0 200 i
* 10.0.2.96/28    172.16.11.1        0      0 1 200 i
*> 10.0.2.96/28    172.16.100.22      0      0 200 i
* 10.0.2.112/28   172.16.11.1        0      0 1 200 i
*> 10.0.2.112/28   172.16.100.22      0      0 200 i
* 10.0.2.128/28   172.16.11.1        0      0 1 200 i
*> 10.0.2.128/28   172.16.100.22      0      0 200 i
```

ISP1 received two routes to 10.0.0.0/24 networks:

- The first route is received from ISP2 (next hop 172.16.100.22), with the AS path 200 1.
- The second route is received from R1 (next hop 172.16.11.1), with the AS path 1.

Since weight and local preference attributes are left on default values, ISP1 chooses the path with shortest AS path length as the best (the route that is received from R1).

If you want to influence the path from AS 100 to 10.0.0.0/24 networks in AS 1, you have to configure AS path prepending.

Configuring AS-Path Prepending

You can configure manual manipulation of the AS path attribute (prepending) using a route map with the **set as-path prepend** command. The route map is used to prepend the specified AS numbers to outgoing EBGp route updates that are matched with the match condition, as specified in the route map. 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.

```
router(config)#
route-map name permit sequence
```

```
match condition
set as-path prepend as-number [ as-number ... ]
```

- Prepends the specified AS number sequence to the routes matched by the route map entry.
- AS numbers prepended to the AS path from the BGP table; the AS number of the sender always prepended to the end.

```
router(config-router)# neighbor address route-map name out
```

- Applies the route map to outgoing updates sent to the specified BGP neighbor.

You can also use the route-map to select only a subset of routes that should have their AS path manually manipulated. Use the **set as-path prepend** command with the appropriate **route-map permit** statement.



Changing an outgoing route map affects only the BGP updates that are sent after the change. To propagate the new and longer AS path with all announced routes, the routes must all be resent by the router. To do this, use the privileged EXEC command **clear ip bgp** with the **soft out** qualifier.

Step 2

You will now configure the AS path prepending on the R1 router. First configure route-map on the R1 router.

Name the route map "PREPEND." Select all BGP routes and prepend five copies of "1" to the existing AS path attribute that is attached to each route. This will make sure that all traffic from AS 100 to 10.0.0.0/24 networks in AS1 will flow through ISP2 router in AS 200.

```
R1(config)# route-map PREPEND permit 10
R1(config-route-map)# set as-path prepend 1 1 1 1 1
```

The lack of match conditions in the route-map indicates that all routes are matched. The AS path-string "1 1 1 1 1" will be prepended to the existing AS path.

Step 3

On the R1 router, apply the "PREPEND" route map to all outgoing updates to the ISP1 EBGp neighbor.

```
R1(config)# router bgp 1
R1(config-router)# neighbor 172.16.11.11 route-map PREPEND out
```

AS 100 will now receive a route to network 10.0.0.0/24 from R1 with an AS path containing the AS number 1 six times (1 1 1 1 1 1).

Step 4

On the R1 router, reestablish BGP sessions to enforce the application of AS path prepending.

To reestablish BGP session, use the **clear ip bgp** command.

```
R1# clear ip bgp *
```

Step 5

On the ISP1 router, verify new best path to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 24, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
```

RPKI validation codes: V valid, I invalid, N Not found

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/28	172.16.100.22			0	200 1 i
*	172.16.11.1			0	1 1 1 1 1 1 i
*> 10.0.0.16/28	172.16.100.22			0	200 1 i
*	172.16.11.1			0	1 1 1 1 1 1 i
*> 10.0.0.32/28	172.16.100.22			0	200 1 i
*	172.16.11.1			0	1 1 1 1 1 1 i
*> 10.0.0.48/28	172.16.100.22			0	200 1 i
*	172.16.11.1			0	1 1 1 1 1 1 i
*> 10.0.0.64/28	172.16.100.22			0	200 1 i
*	172.16.11.1			0	1 1 1 1 1 1 i
*> 10.0.1.0/28	0.0.0.0	0		32768	i
*> 10.0.1.16/28	0.0.0.0	0		32768	i
*> 10.0.1.32/28	0.0.0.0	0		32768	i
*> 10.0.1.48/28	0.0.0.0	0		32768	i
*> 10.0.1.64/28	0.0.0.0	0		32768	i
*> 10.0.1.80/28	0.0.0.0	0		32768	i
Network	Next Hop	Metric	LocPrf	Weight	Path
* 10.0.2.0/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.16/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.32/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.48/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.64/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.80/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.96/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.112/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i
* 10.0.2.128/28	172.16.11.1			0	1 1 1 1 1 1 200 i
*>	172.16.100.22	0		0	200 i

ISP1 received two routes to 10.0.0.0/24 networks:

- The first route is received from ISP2 (next hop 172.16.100.22), with the AS path 200 1.
- The second route is received from R1 (next hop 172.16.11.1), with the AS path 1 1 1 1 1 1.

You can see now, that the received networks have the "1 1 1 1" AS path that is prepended to the original AS path. Consequently, the received route from R1 has a longer AS path than the route received from ISP2. Consequently, as weight and local preference value are left on default, ISP1 chooses the path that is received from ISP2 as the best (shortest AS path length).

Note: 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:

- Output of the **debug ip bgp updates** command does not display the prepended paths, because the route map doing the prepending is applied afterward.
- Output of **show route-map** command displays the configuration details of a route map. However, there is no indication of how many routes have been matched by a route-map statement.

A better place for observing AS path prepending is on the router receiving the BGP update that contains the prepended AS path. You can also 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

Service providers 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 of the rest of the Internet cause the service provider to implement AS path filters on incoming updates.

The network administrator of the service provider can 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 is a better solution. This access list can then be uniformly applied to all incoming routes from all customers.



When you configure AS path prepending, you have to agree with the service provider so their AS path filters will be modified to support AS path prepending.

Step 6

Imagine now for a moment that you are a service provider and configure AS path filters to prevent AS path prepending. **NOTE:** A customer normally does not configure the [ISP](#) side. The configuration on the side of ISP is done by their network administrator.

On the ISP1 router, configure AS path filter, a filter list, that will not support AS path prepending but will allow only AS paths with a length of one AS number. Apply this filter list to all incoming updates from R1 EBGp neighbor.

```
ISP1(config)# ip as-path access-list 10 permit ^[0-9]+$
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.11.1 filter-list 10 in
```

A filter list allows only AS paths with a length of one AS number—this number can be any number. Consequently, all routes with a prepended AS path are filtered out.

Step 7

On the ISP1 router, reestablish BGP sessions to enforce the application of filter list.

To reestablish BGP session, use the **clear ip bgp *** command.

```
ISP1# clear ip bgp *
```

Step 8

On the ISP1 router, verify received routes to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 20, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/28	172.16.100.22			0	200 1 i
*> 10.0.0.16/28	172.16.100.22			0	200 1 i
*> 10.0.0.32/28	172.16.100.22			0	200 1 i
*> 10.0.0.48/28	172.16.100.22			0	200 1 i
*> 10.0.0.64/28	172.16.100.22			0	200 1 i
*> 10.0.1.0/28	0.0.0.0			32768	i
*> 10.0.1.16/28	0.0.0.0			32768	i
*> 10.0.1.32/28	0.0.0.0			32768	i
*> 10.0.1.48/28	0.0.0.0			32768	i
*> 10.0.1.64/28	0.0.0.0			32768	i
*> 10.0.1.80/28	0.0.0.0			32768	i
*> 10.0.2.0/28	172.16.100.22			0	200 i
*> 10.0.2.16/28	172.16.100.22			0	200 i
*> 10.0.2.32/28	172.16.100.22			0	200 i
*> 10.0.2.48/28	172.16.100.22			0	200 i
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.2.64/28	172.16.100.22			0	200 i
*> 10.0.2.80/28	172.16.100.22			0	200 i
*> 10.0.2.96/28	172.16.100.22			0	200 i
*> 10.0.2.112/28	172.16.100.22			0	200 i
*> 10.0.2.128/28	172.16.100.22			0	200 i

ISP1 now received only one route to 10.0.0.0/24 networks, the one from ISP2 (next hop 172.16.100.22). Since R1 is announcing the networks 10.0.0.0/24 with a prepended AS path, the filter-list for incoming routes to AS 100 filtered those routes out. This filtering results in a situation where the networks 10.0.0.0/24 are not reachable over the link between R1 and ISP1 (AS 1 and AS 100). Therefore, the backup function is not available.

Networks 10.0.0.0/24 are, however, still reachable via the path going through AS 200.

Step 9

If you want to have a backup path available to 10.0.0.0/24 networks, you have to meet an agreement with a customer and allow AS path prepending in your filters.

On the ISP1 router, modify incoming filter to allow one or multiple copies of the same AS number. Configure a filter for all routes that are received directly from AS 1.

```
ISP1(config)# ip as-path access-list 10 permit ^1(_1)*$
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.11.1 filter-list 10 in
```

You configured an individual filter for all routes that are received directly from AS 1. The AS path is required to start with 1, then multiple copies of 1 may follow it.

If you had more customers, they would all require an individual filter list, because the customer is explicitly indicated in the regular expression. An alternative would be to implement the AS path filter that allows any AS path containing one or multiple copies of the same AS number. For example using `^([0-9]+)(_1)*$` filter, that matches any AS path beginning with any AS number and continues with no or multiple repetitions of the same AS number.

Step 10

On the ISP1 router, reestablish BGP sessions to enforce the application of new filter list.

To re-establish BGP session, use the **clear ip bgp** command.

```
ISP1# clear ip bgp *
```

Step 11

On the ISP1 router, verify new received routes to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 20, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28     172.16.100.22      0         0 200 1 i
*                  172.16.11.1        0         0 1 1 1 1 1 1 i
*> 10.0.0.16/28    172.16.100.22      0         0 200 1 i
*                  172.16.11.1        0         0 1 1 1 1 1 1 i
*> 10.0.0.32/28    172.16.100.22      0         0 200 1 i
*                  172.16.11.1        0         0 1 1 1 1 1 1 i
*> 10.0.0.48/28    172.16.100.22      0         0 200 1 i
*                  172.16.11.1        0         0 1 1 1 1 1 1 i
*> 10.0.0.64/28    172.16.100.22      0         0 200 1 i
*                  172.16.11.1        0         0 1 1 1 1 1 1 i
*> 10.0.1.0/28     0.0.0.0            0         0 32768 i
*> 10.0.1.16/28    0.0.0.0            0         0 32768 i
*> 10.0.1.32/28    0.0.0.0            0         0 32768 i
*> 10.0.1.48/28    0.0.0.0            0         0 32768 i
*> 10.0.1.64/28    0.0.0.0            0         0 32768 i
*> 10.0.1.80/28    0.0.0.0            0         0 32768 i
   Network        Next Hop        Metric LocPrf Weight Path
*> 10.0.2.0/28     172.16.100.22      0         0 200 i
*> 10.0.2.16/28    172.16.100.22      0         0 200 i
*> 10.0.2.32/28    172.16.100.22      0         0 200 i
*> 10.0.2.48/28    172.16.100.22      0         0 200 i
*> 10.0.2.64/28    172.16.100.22      0         0 200 i
*> 10.0.2.80/28    172.16.100.22      0         0 200 i
*> 10.0.2.96/28    172.16.100.22      0         0 200 i
*> 10.0.2.112/28   172.16.100.22      0         0 200 i
*> 10.0.2.128/28   172.16.100.22      0         0 200 i
```

ISP1 again received two routes to 10.0.0.0/24 networks:

- The first route is received from ISP2 (next hop 172.16.100.22), with the AS path 200 1.
- The second route is received from R1 (next hop 172.16.11.1), with the AS path 1 1 1 1 1 1.

As weight and local preference value are left on default, ISP1 chooses the path that is received from ISP2 as the best based on the shortest AS path length.

BGP Hide Local-Autonomous System

Changing the [AS](#) number may be necessary when two separate [BGP](#) networks are combined under a single AS, a situation that typically occurs when one [ISP](#) purchases another ISP. However changing of the AS number in a BGP network can be difficult. During the transition, [IBGP](#) peers will reject external routes from peers with a local AS number in the AS number path to prevent routing loops. The BGP Hide Local-Autonomous System feature allows you to transparently change the AS number for the entire BGP. It also ensures that routes can be propagated throughout the AS, while the AS number transition is incomplete.

- Allows you to transparently change the AS number for the entire BGP network.
- Ensures that routes can be propagated throughout the AS.
- **neighbor local-as** command in address family or router configuration mode allows customization of the AS number for EBGp peer groupings.

To allow customization of the AS number for [EBGP](#) peer groupings, use the **neighbor local-as** command in address family or router configuration mode. To disable this function, use the **no** form of this command.

```
neighbor {ip-address | peer-group-name} local-as as-number [no-prepend]
```

Syntax Description

Parameter	Description
<i>ip-address</i>	IP address of the local BGP-speaking neighbor
<i>peer-group-name</i>	Name of a BGP peer group.
<i>as-number</i>	Valid AS number from 1 to 65535. Do not specify the AS number to which the neighbor belongs.
no-prepend	Configures the router to not prepend the local AS number to any routes received from an external peer.

The **neighbor local-as** command is used initially to configure BGP peers to support two local AS numbers to maintain peering between two separate BGP networks. This configuration allows the ISP to immediately make the transition without any impact on existing customer configurations. When the customer configurations have been updated, the next step is to complete the transition from the old AS number to the new AS number.

When the **neighbor local-as** command is configured on a BGP peer, the local AS number is automatically prepended to all routes that are learned from EBGp peers by default. This behavior, however, makes changing the AS number for a service provider or large BGP network difficult. Routes with the prepended AS number will be rejected by IBGP peers that are configured with the same AS number.

For example, if you configure an IBGP peer with the **neighbor 10.0.0.2 local-as 20** statement, all routes that are learned from the 10.0.0.2 external peer will automatically have the AS number 20 prepended. Internal routers that are configured with the AS number 20 will detect these routes as routing loops and reject them. This behavior requires you to change the AS number for all IBGP peers at the same time.

Summary

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

- If you do not configure the preferred path for incoming (return) traffic, the likely result is an asymmetrical traffic flow as well as suboptimal performance of the return traffic.
- AS path prepending is performed on outgoing EBGp updates over the undesired return path. Configure it using a route-map with the **set as-path prepend** command.
- Monitor AS path prepending on the router that is receiving the prepended routes because the prepended path will not be visible on the prepending router.
- If you are a service provider with customers that use AS path prepending, you must create new AS path filters to accommodate AS path lengths greater than one AS number.
- The BGP Hide Local-Autonomous System feature allows you to transparently change the AS number for the entire BGP network. It also ensures that routes can be propagated throughout the AS, while the AS number transition is incomplete.

Understanding BGP Multi-Exit Discriminators

Overview

When connections to multiple providers are required, it is important that [BGP](#) selects the optimum route for traffic to use. It is equally important that the selected return path is the optimum return path into the [AS](#). 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 a tool for administrators to influence route selection, the [MED](#) attribute.

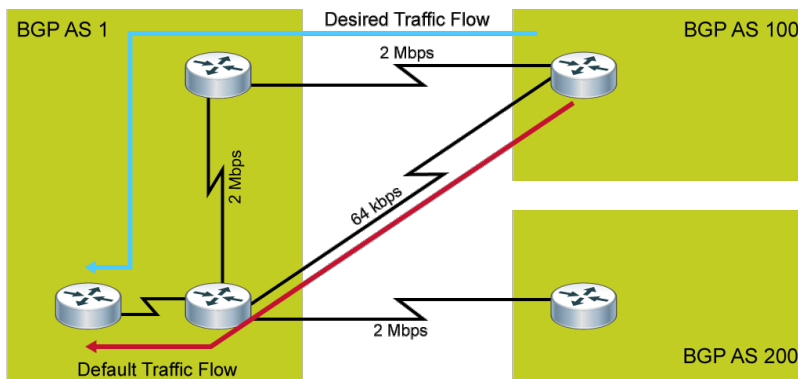
In this lesson, you will learn how to influence BGP route selection by setting the BGP MED attribute of outgoing BGP routes. Two methods that are used to set the MED attribute, the default MED and route maps, are discussed. In addition, you will also learn the basic MED attribute configuration, advanced commands to manipulate MED properties, as well as how to monitor and troubleshoot the BGP table to verify correct MED configuration and to properly influence path selection.

Upon completing this lesson, you will be able to:

- Describe how the MED can be used to facilitate proper return path selection
- Explain how the value of the MED attribute changes inside a BGP AS and between different BGP autonomous systems
- Identify the Cisco IOS command that is required to configure BGP MED
- Describe how to configure and monitor BGP MED
- Identify the Cisco IOS commands that are required to troubleshoot the BGP MED configurations
- Identify the Cisco IOS commands that are required to configure advanced MED features on Cisco routers

Selecting the Proper Return Path

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.



Q: How can you make sure that the return traffic takes the right path?

More complex half of the problem is how to influence neighboring autonomous systems to choose the correct return path back into the AS.

Multi-Exit Discriminator

The MED attribute is a hint to external neighbors about the preferred path into an AS when multiple entry points exist.

You can use the MED to influence path selection in neighbor autonomous systems.

- An AS can specify its preferred entry point using the MED in outgoing EGP updates.
- The MED is not propagated outside of a receiving AS.
- The default value of the MED attribute is 0.
- The MED is called the "metric" in Cisco IOS software.
- The MED is a "weak" metric.
- A **lower** MED value means more preferred.

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.

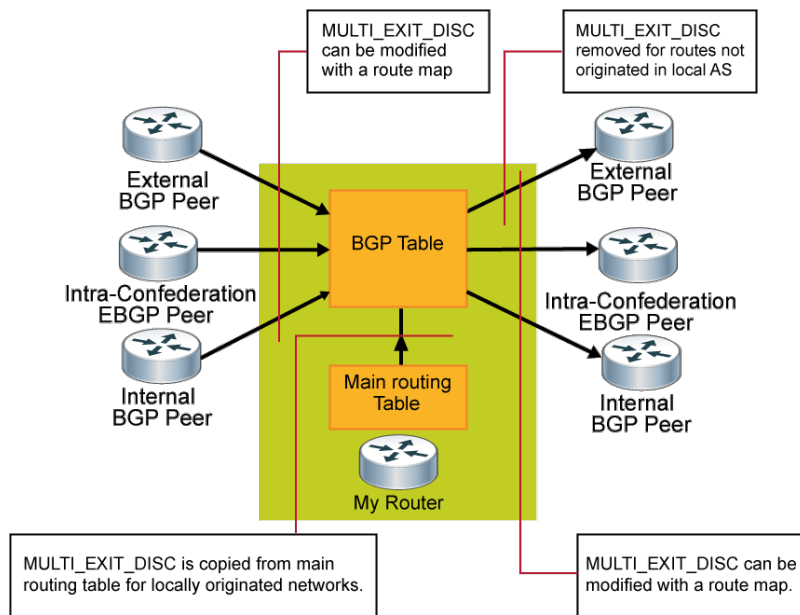
The MED attribute is considered a "weak" metric. In contrast with weight and local preference, a router will prefer a path with the smallest MED value. But only if the weight, local preference, AS path, and origin code attributes are equal. Using the MED may not yield the expected result if the neighboring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



The term that is used in Cisco IOS software for MED is "metric." The term "metric" also applies to the set command that is used in route maps as well as all **show** and **debug** commands.

MED Propagation in a BGP Network

The figure shows how the value of the [MED](#) attribute is assigned, depending upon the routing information source.



You must configure a route map 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 MED

The **MED** is not a mandatory attribute, and there no MED attribute is attached to a route by default. The only exception is if the router is originating networks that have an exact match in the routing table (through the **network** command or through redistribution). In that case, the router uses the metric in the routing table as the MED attribute value.

You have two options, either you change the default MED or you change the MED with route map.

- Option 1:

```
router(config-router)# default-metric number
```

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

- Option 2:

```
router(config)#  
route-map name permit sequence  
  match condition  
  set metric value
```

- Changes MED for routes that are matched by the route map entry

```
router(config-router)# neighbor address route-map name in | out
```

- Applies a route map to incoming updates from a specified neighbor or to outgoing updates to a specified neighbor.
- Per-neighbor MED configured by using a route map with no match condition.

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

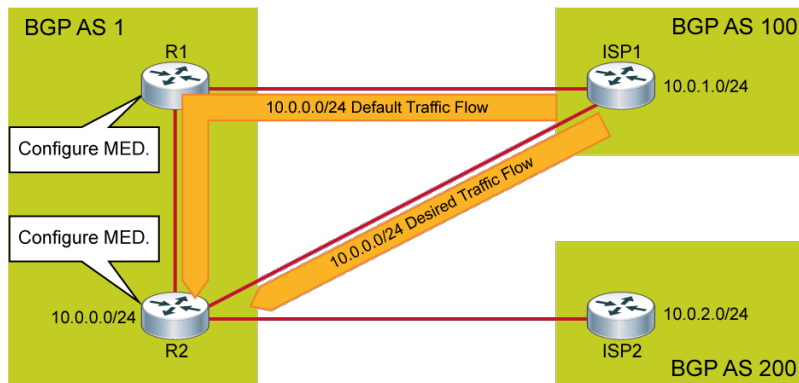
To set the default metric (MED) value for BGP routes, use the **default-metric** *number* command in router configuration mode. *number* is a default metric value that is appropriate for the specified routing protocol. To return to the default state, use the **no** form of this command.

You can also use a route map to set the MED on incoming or outgoing updates. Use the **set metric** command within route map configuration mode to set the MED attribute.

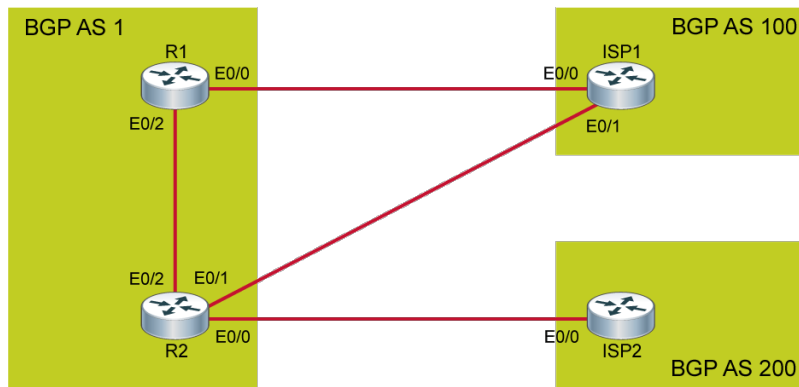
Discovery 15: Configure MED

Overview

Through this discovery, you will learn how to configure [MED](#) using route maps to influence the return path from [AS 100](#) to 10.0.0.0/24 network in AS 1.



Topology



Job Aids

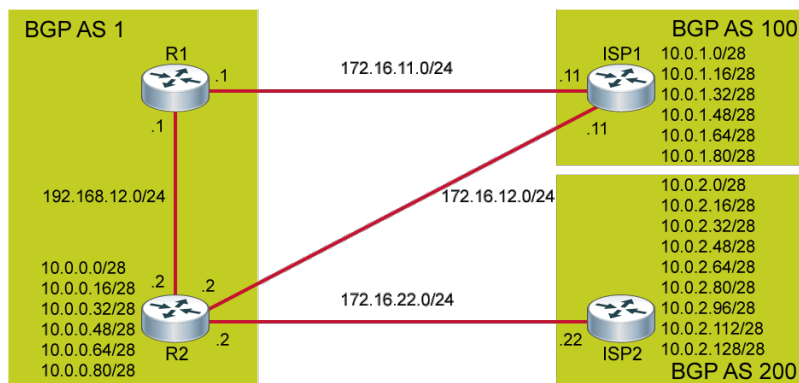


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/1	172.16.12.2/24	Connection to ISP1
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/0	172.16.11.11/24	Connection to R1
ISP1	Ethernet 0/1	172.16.12.11/24	Connection to R2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured:

- [EBGP](#)
 - R1 to ISP1
 - R2 to ISP1
 - R2 to ISP2

- [IBGP](#)
 - R1 to R2

Configure MED

Discovery Steps

Step 1

On the ISP1 router, re-establish the BGP session.

To re-establish BGP session, use the **clear ip bgp *** command.

```
ISP1# clear ip bgp *
```

Note: Because of the limitation of IOL, you have to reestablish the BGP session on ISP1 to get the same results as described in the next step of the lab.

Step 2

On the ISP1 router, verify the initial best path to reach 10.0.0.0/24 network.

```
ISP1# show ip bgp
BGP table version is 20, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*> 10.0.0.0/28     172.16.11.1              0 1 i
*                  172.16.12.2              0 1 i
*> 10.0.0.16/28    172.16.11.1              0 1 i
*                  172.16.12.2              0 1 i
*> 10.0.0.32/28    172.16.11.1              0 1 i
*                  172.16.12.2              0 1 i
*> 10.0.0.48/28    172.16.11.1              0 1 i
*                  172.16.12.2              0 1 i
*> 10.0.0.64/28    172.16.11.1              0 1 i
*                  172.16.12.2              0 1 i
*> 10.0.1.0/28     0.0.0.0                 0 32768 i
*> 10.0.1.16/28    0.0.0.0                 0 32768 i
*> 10.0.1.32/28    0.0.0.0                 0 32768 i
*> 10.0.1.48/28    0.0.0.0                 0 32768 i
*> 10.0.1.64/28    0.0.0.0                 0 32768 i
*> 10.0.1.80/28    0.0.0.0                 0 32768 i
   Network        Next Hop        Metric LocPrf Weight Path
*> 10.0.2.0/28     172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.16/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.32/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.48/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.64/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.80/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.96/28    172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.112/28   172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
*> 10.0.2.128/28   172.16.11.1              0 1 200 i
*                  172.16.12.2              0 1 200 i
```

ISP1 has received 10.0.0.0/24 networks from two different neighbors:

- The first path was received from R1 (next hop 172.16.11.1).
- The second path was received from R2 (next hop 172.16.12.2).

ISP1 chooses the first path (the one received from R1) as the best path.

Step 3

On the ISP1 router, check why was the route that was received from R1 chosen as the best path to 10.0.0.0/24 networks.

```
ISP1# show ip bgp
```

```

BGP table version is 20, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/28	172.16.11.1			0 1	i
* 10.0.0.0/28	172.16.12.2	0		0 1	i
*> 10.0.0.16/28	172.16.11.1			0 1	i
* 10.0.0.16/28	172.16.12.2	0		0 1	i
*> 10.0.0.32/28	172.16.11.1			0 1	i
* 10.0.0.32/28	172.16.12.2	0		0 1	i
*> 10.0.0.48/28	172.16.11.1			0 1	i
* 10.0.0.48/28	172.16.12.2	0		0 1	i
*> 10.0.0.64/28	172.16.11.1			0 1	i
* 10.0.0.64/28	172.16.12.2	0		0 1	i

<... output omitted ...>

If you follow the route selection criteria, you can determine that:

- The weight is not configured for any neighbor.
- The local preference is not configured for any neighbor.
- Routes do not originate locally.
- Both routes have the same length of AS path.
- Both routes are [EBGP](#) routes.

Since both paths are EBGP paths, the oldest (most stable) path is preferred. Next thing that you have to verify is the duration of session between ISP1 and R1, and ISP1 and R2.

```

ISP1# show bgp summary
BGP router identifier 10.0.1.81, local AS number 100
BGP table version is 20, main routing table version 20
19 network entries using 2812 bytes of memory
32 path entries using 2048 bytes of memory
4/3 BGP path/bestpath attribute entries using 544 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 5452 total bytes of memory
BGP activity 95/76 prefixes, 160/128 paths, scan interval 60 secs

```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
172.16.11.1	4	1	25	25	20	0	0	00:17:40	13
172.16.12.2	4	1	28	25	20	0	0	00:17:40	13

The duration of both session is the same. So obviously, the router-ID breaks the tie—the path from router with the lower BGP router-ID is preferred.

```

ISP1# show ip bgp neighbor 172.16.11.1
BGP neighbor is 172.16.11.1, remote AS 1, external link
  BGP version 4, remote router ID 10.0.7.1
  BGP state = Established, up for 00:21:49
  Last read 00:00:42, last write 00:00:54, hold time is 180, keepalive
<... output omitted ...>

```

```

ISP1# show ip bgp neighbor 172.16.12.2
BGP neighbor is 172.16.12.2, remote AS 1, external link
  BGP version 4, remote router ID 10.0.7.17
  BGP state = Established, up for 00:23:40
  Last read 00:00:44, last write 00:00:50, hold time is 180, keepalive
<... output omitted ...>

```

The router-ID of R1 is lower than the router-ID of R2, so the route received from R1 is preferred.

Step 4

On the customer side, influence the returning traffic flow to 10.0.0.0/24 networks, so the ISP1 router will take the path via R2 router instead the default one. Reconfigure MED using route maps.

Configure route map "MED" on both routers, R1 and R2, to prefer the lower link

to AS 1.

```
R1(config)# route-map MED
R1(config-route-map)# set metric 500
```

```
R2(config)# route-map MED
R2(config-route-map)# set metric 100
```

Lower MED value is preferred when choosing the best path. If you want R2 to be chosen for the returning traffic, it has to be configured with lower MED.

Step 5

On the R1 and R2 routers, apply route maps for ISP1 neighbor in the outbound direction.

```
R1(config)# router bgp 1
R1(config-router)# neighbor 172.16.11.11 route-map MED out
```

```
R2(config)# router bgp 1
R2(config-router)# neighbor 172.16.12.11 route-map MED out
```

Do not forget that the solution of influencing the returning traffic flow, of course, relies on the neighboring AS not changing the weight, local preference, AS path, or origin code attributes in updates that it receives from AS 1.

Step 6

On the ISP1 router, reestablish BGP sessions to enforce the application of MED. To reestablish BGP session, use the **clear ip bgp** command.

```
ISP1# clear ip bgp *
```

Step 7

On the ISP1 router, verify new best path to reach 10.0.0.0/24 network.

```
ISP1# show ip bgp
BGP table version is 46, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*    10.0.0.0/28    172.16.11.1         500             0 1 i
*>    10.0.0.0/28    172.16.12.2         100             0 1 i
*    10.0.0.16/28   172.16.11.1         500             0 1 i
*>    10.0.0.16/28   172.16.12.2         100             0 1 i
*    10.0.0.32/28   172.16.11.1         500             0 1 i
*>    10.0.0.32/28   172.16.12.2         100             0 1 i
*    10.0.0.48/28   172.16.11.1         500             0 1 i
*>    10.0.0.48/28   172.16.12.2         100             0 1 i
*    10.0.0.64/28   172.16.11.1         500             0 1 i
*>    10.0.0.64/28   172.16.12.2         100             0 1 i
*>    10.0.1.0/28    0.0.0.0              0            32768 i
*>    10.0.1.16/28    0.0.0.0              0            32768 i
*>    10.0.1.32/28    0.0.0.0              0            32768 i
*>    10.0.1.48/28    0.0.0.0              0            32768 i
*>    10.0.1.64/28    0.0.0.0              0            32768 i
*>    10.0.1.80/28    0.0.0.0              0            32768 i
   Network        Next Hop        Metric LocPrf Weight Path
*    10.0.2.0/28    172.16.11.1         500             0 1 200 i
*>    10.0.2.0/28    172.16.12.2         100             0 1 200 i
*    10.0.2.16/28   172.16.11.1         500             0 1 200 i
*>    10.0.2.16/28   172.16.12.2         100             0 1 200 i
*    10.0.2.32/28   172.16.11.1         500             0 1 200 i
*>    10.0.2.32/28   172.16.12.2         100             0 1 200 i
*    10.0.2.48/28   172.16.11.1         500             0 1 200 i
```

*>		172.16.12.2	100	0 1 200 i
*	10.0.2.64/28	172.16.11.1	500	0 1 200 i
*>		172.16.12.2	100	0 1 200 i
*	10.0.2.80/28	172.16.11.1	500	0 1 200 i
*>		172.16.12.2	100	0 1 200 i
*	10.0.2.96/28	172.16.11.1	500	0 1 200 i
*>		172.16.12.2	100	0 1 200 i
*	10.0.2.112/28	172.16.11.1	500	0 1 200 i
*>		172.16.12.2	100	0 1 200 i
*	10.0.2.128/28	172.16.11.1	500	0 1 200 i
*>		172.16.12.2	100	0 1 200 i

ISP1 has received 10.0.0.0/24 networks from two different neighbors:

- The first path was received from R1 (next hop 172.16.11.1), with the MED set to 500.
- The second path was received from R2 (next hop 172.16.12.2), with the MED set to 100.

ISP1 chooses the path that was received from R2 as the best path, since it has lower MED configured.

Note that MED is displayed in the **show ip bgp** printout as the metric field. All BGP-related **show** and **debug** commands display the value of the MED attribute. If the MED is not configured, the metric field is left blank. When looking at detailed information for a specific network (via the **show ip bgp prefix** command), you will see the MED only if the attribute exists.

Troubleshooting the MED

If debugging is necessary to troubleshoot a problem, the [MED](#), among other attributes, is displayed in the outputs.

- MED sent to a neighbor (after the outgoing route map) is displayed in debugging outputs.

```
R1# debug ip bgp updates
BGP updates debugging is on for address family: IPv4 Unicast
R1# clear ip bgp 172.16.11.1 soft out
R1#
*Mar  9 01:58:12.497: BGP(0): (base) 172.16.11.11 send UPDATE (format) 10.0.0.0/28, next 172.16.11.1, metric 500, path Local
*Mar  9 01:58:12.497: BGP(0): (base) 172.16.11.11 send UPDATE (format) 10.0.0.16/28, next 172.16.11.1, metric 500, path Local
*Mar  9 01:58:12.497: BGP(0): (base) 172.16.11.11 send UPDATE (format) 10.0.0.32/28, next 172.16.11.1, metric 500, path Local
```

The MED attribute was set with an outgoing route map. This MED attribute ("metric 500") is then sent to the neighbor in the update.

- MED received from a neighbor is displayed in debugging outputs.

```
ISP1# debug ip bgp updates
BGP updates debugging is on for address family: IPv4 Unicast
ISP1# clear ip bgp 172.16.11.1
ISP1#
*Mar  9 02:31:35.697: BGP(0): 172.16.11.1 rcvd UPDATE w/ attr: nexthop 172.16.11.1, origin i, metric 500, merged path 1, AS_PATH
*Mar  9 02:31:35.697: BGP(0): 172.16.11.1 rcvd 10.0.0.0/28
*Mar  9 02:31:35.697: BGP(0): 172.16.11.1 rcvd 10.0.0.16/28
*Mar  9 02:31:35.698: BGP(0): 172.16.11.1 rcvd 10.0.0.32/28
```

This debugging example shows the received updates for 10.0.0.0/28, 10.0.0.0/16, and 10.0.0.32/28 networks. The MED attribute ("metric 500") for all three networks is shown in the received update. The MED was set on the neighboring router.

- Original MED received from a neighbor (before the incoming route map processing) is displayed in **show ip bgp neighbor address received-routes**.

```
ISP1# show ip bgp neighbor 172.16.11.1 received-routes
BGP table version is 59, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*   10.0.0.0/28      172.16.11.1           500             0 1 i
*   10.0.0.16/28     172.16.11.1           500             0 1 i
*   10.0.0.32/28     172.16.11.1           500             0 1 i

Total number of prefixes 3
```

You can change the MED of the received route on the receiving router (on ISP1 for this example). To see the original MED, you need to enable soft reconfiguration on the router. The **show ip bgp neighbor address received-routes** command displays the original updates before any filters or route maps have filtered or changed them.

In the example, the ISP1 router received three networks from 172.16.11.1 neighbor—10.0.0.0/28, 10.0.0.16/28, and 10.0.0.32/28. All three networks have the metric set to 500.

- Both, original route and modified route, are displayed with a route map when inbound soft reconfiguration is configured.

```
ISP1# show ip bgp 10.0.0.0/28
BGP routing table entry for 10.0.0.0/28, version 56
Paths: (3 available, best #1, table default)
  Advertised to update-groups:
    5
  Refresh Epoch 2
    1
```

```
172.16.11.1 from 172.16.11.1 (10.0.7.1)
  Origin IGP, metric 50, localpref 100, valid, external, best
Refresh Epoch 2
  1, (received-only)
172.16.11.1 from 172.16.11.1 (10.0.7.1)
  Origin IGP, metric 500, localpref 100, valid, external
```

- Only the modified route is displayed in the BGP table.

```
ISP1# show ip bgp
BGP table version is 163, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop          Metric LocPrf Weight Path
*>  10.0.0.0/28      172.16.11.1           50             0 1 i
*>  10.0.0.16/28     172.16.11.1           50             0 1 i
*>  10.0.0.32/28     172.16.11.1           50             0 1 i
```

If you configured the route map to influence the MED of received route, you can see the original updates to the MED attribute by using the **show ip bgp prefix** command. Soft reconfiguration has to be enabled. The original versions are marked with the **received-only** keyword and follow the version that is in the global [BGP](#) table.

In this example, the received update had MED attribute set to 500, but a value of 50 was later applied through an inbound route map.

Advanced MED Configuration

Several rules exist on when and how you should use the [MED](#) attribute.

By default, the MED is considered only during selection of routes from the same AS.

```
router(config-router)# bgp always-compare-med
```

- With **always-compare-med**, the MED is also considered for routes coming from a different AS.

If the MED is not attached to a BGP route, it is interpreted as value 0, and thus as the best metric.

```
router(config-router)# bgp bestpath med missing-med-worst
```

- With this command, a missing MED is interpreted as infinity (worst).
1. You should use the MED in the route selection process only if both (all) paths come from the same [AS](#). Use the **bgp always-compare-med** 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.
 2. According to a [BGP](#) standard describing MED, you should regard a missing MED attribute as an infinite value. Cisco IOS software, on the other hand, regards a missing MED attribute as having a value of 0. Use the **bgp bestpath med missing-med-worst** command when combining equipment from different vendors. An even better solution is to make sure that every update carries a MED attribute.

By default, the MED is considered only during selection of routes from the same AS, which does not include intra-confederation autonomous systems.

```
router(config-router)# bgp bestpath med confed
```

- Use this command to allow routers to compare paths that are learned from confederation peers.

```
router(config-router)# bgp deterministic-med
```

- This command changes the BGP route selection procedure to a deterministic but slower one.
3. You must use the **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.
 4. 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.



Cisco recommends enabling the **bgp deterministic-med** command in all new network rollouts. For existing networks, you must deploy the command either on all routers at the same time or incrementally, with care to avoid possible [IBGP](#) routing loops.

Advanced MED Configuration Example

The following example demonstrates how the **bgp deterministic-med** and **bgp always-compare-med** commands can influence MED-based path selection. Consider the following BGP routes for network 172.16.0.0/16 in the order that they are received:

```
entry 1: AS(PATH) 65500, med 150, external, rid 192.168.13.1  
entry 2: AS(PATH) 65100, med 200, external, rid 1.1.1.1  
entry 3: AS(PATH) 65500, med 100, internal, rid 192.168.8.4
```

BGP compares multiple routes to a single destination in



pairs, starting with the newest entry and moving toward the oldest entry (starting at the top of the list and moving down). For example, entry 1 and entry 2 are compared. The better of these two is then compared to entry 3, and so on.

In the case where both commands are disabled, BGP compares entry 1 and entry 2. Entry 2 is chosen as the better one because it has a lower router-ID. The MED is not checked because the paths are from a different neighbor AS. Next, entry 2 is compared to entry 3. BGP chooses entry 2 as the best path because it is external.

In the case where the **bgp deterministic-med** command has been disabled and the **bgp always-compare-med** command has been enabled, BGP compares entry 1 to entry 2. These entries are from different autonomous systems, but because the **bgp always-compare-med** command is enabled, the MED is used in the comparison. Entry 1 is the better of these two entries because it has a lower MED value. Next, BGP compares entry 1 to entry 3. The MED is checked again because the entries are now from the same AS. BGP chooses entry 3 as the best path.

In the case where the **bgp deterministic-med** command has been enabled and the **bgp always-compare-med** command has been disabled, BGP groups routes from the same AS. Then it compares the best entries of each group. The BGP table looks like the following:

```
entry 1: AS(PATH) 65100, med 200, external, rid 1.1.1.1
entry 2: AS(PATH) 65500, med 100, internal, rid 192.168.8.4
entry 3: AS(PATH) 65500, med 150, external, rid 192.168.13.1
```

There is a group for AS 65100 and a group for AS 65500. BGP compares the best entries for each group. Entry 1 is the best of its group because it is the only route from AS 100. BGP compares entry 1 to the best of group AS 65500, entry 2 (because it has the lowest MED). Because the two entries are not from the same neighbor AS, the MED is not considered in the comparison. The [EBGP](#) route wins over the IBGP route, making entry 1 the best route.

If the **bgp always-compare-med** command were also enabled, BGP would have taken the MED into account for the last comparison and have selected entry 2 as the best path.

Summary

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

- 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.
- There is no MED attribute that is attached to a route by default.
 - You can set the default metric value (MED) for BGP routes, using the `default-metric` command in router configuration mode.
 - You can use a route map to set an arbitrary MED value to sent or received routes.
- You can configure advanced MED parameters to modify the default MED behaviors.
 - Use the **`bgp always-compare-med`** to force the router to compare the MED even if paths came from different autonomous systems.
 - Use the **`bgp bestpath med confed`** command when you use the MED within a confederation to influence the route selection process

Addressing BGP Communities

Overview

When connections to multiple [ISPs](#) are required, it is important that [BGP](#) select the optimum route for traffic to use. It is equally important that the selected return path is the optimum return path into the [AS](#). BGP provides tools for administrators to influence route selection and BGP community attribute is one such tool.

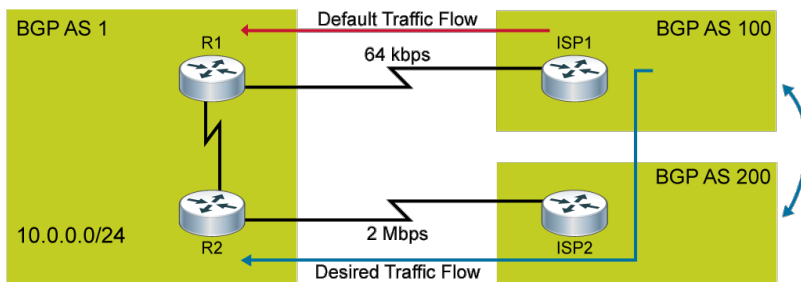
You will learn how to influence BGP route selection by setting the BGP community attribute on outgoing BGP routes to facilitate proper return path selection. You will also learn about the configuration details of BGP communities.

Upon completing this lesson, you will be able to:

- Describe the issues of return path selection for multihomed customers and why you cannot use the BGP attributes of weight, local preference, and MED to solve these issues
- Describe the basic qualities of BGP communities
- Describe how BGP communities facilitate proper return path selection
- List the steps that are required to successfully deploy communities in a BGP-based network
- Identify the Cisco IOS commands that are required to deploy BGP communities
- Describe the function of the BGP Named Community Lists feature
- Describe the function of the BGP Cost Community feature
- Describe the function of the BGP Link Bandwidth feature
- Describe how BGP supports sequenced entries in extended community-lists

Selecting the Proper Return Path

In this example, the customer and the backup service provider would like to avoid [AS](#) path prepending and rely on other [BGP](#) tools to properly route the return traffic over the highest-speed [WAN](#) link.



Q: How do you select the proper return path from AS 100 without using AS path prepending in AS 1?

A: Use local preference in AS 100.

Q: Will the administrator of AS 100 configure it?

A: Unlikely.

Using the [MED](#) to influence the preferred return path is not possible because the MED cannot be propagated across several autonomous systems. AS 100 would, therefore, receive networks from AS 1 directly with the MED attribute but would receive networks without a MED attribute from AS 200. In any case, BGP route selection would be based on the length of the AS path. Even if the MED was present and used the shortest path, the path would still be through the slow 64-kbps link.

The only option for resolving this issue is to use local preference in AS 100. The problem with this solution is that service providers normally do not rush to implement every wish that their customers might have.

In this lesson, you will learn a solution to this case study that uses the transitive optional attribute called "BGP community" in conjunction with local preference.

BGP Communities Overview

BGP communities are attributes that are used to group and filter routes.

Communities are designed to give you the ability to apply policies to large numbers of routes by using match and set clauses in the configuration of route maps.

Community lists are used in this process to identify and filter routes by their common attributes.

- BGP communities are a mean of tagging routes to ensure consistent filtering or route selection policy.
- Any BGP router can tag routes in incoming and outgoing routing updates or when doing redistribution.
- Any BGP router can filter routes in incoming or outgoing updates or select preferred routes based on communities.
- By default, communities are stripped in outgoing BGP updates.

A community is an attribute that is used to tag BGP routes. A router can apply it to any BGP route by using a 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 transitive optional attribute. Its value is a 32-bit number (range 0 to 4,294,967,200).
- Each network in a BGP routing table can be tagged with a set of communities.
- 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.
- Routers that do not support communities pass them along unchanged.

The community attribute is a 32-bit transitive optional BGP attribute that was designed to group destinations and apply routing decision (accept, prefer, redistribute, and so on) according to communities. This way, the community attribute allows easy application of administrative policies. BGP communities provide a mechanism to reduce BGP configuration complexity on a router 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 as follows:

- **no-advertise**: If a router receives an update carrying this community, it will not forward it to any neighbor.
- **no-export**: If a router receives an update carrying this community, it will not propagate it to any external neighbors except to intra-confederation external neighbors. The **no-export** attribute 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 intra-confederation 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.

Defining your own communities

- 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.
- Values of all zeroes and all ones in high-order 16 bits are reserved.
- Cisco IOS parser allows you to specify a 32-bit community value as:
 - [AS-number]:[low-order-16-bits]

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. 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).

You can also use communities internally within an AS (to ensure AS-wide routing policy), in which case the first 16 bits should contain the AS number of the local AS.

Using Communities

- Define administrative policy goals.
- Design filters and route selection policy to achieve administrative goals.
- Define communities that signal individual goals.
- 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.

Designing a [BGP](#) solution around BGP communities usually requires the following steps:

1. Define 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.
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.

Using Communities Example

This example shows how you can define goals and assign communities to them.

- Define administrative policy goal:
 - Solve asymmetrical customer routing problems.
- Design filters and path selection policy to achieve administrative goals:
 - Set local preference of customer routes to 50 for customers using the backup ISP.
- Define communities that signal individual goals:
 - Community 100:17 is used to indicate that the local preference of the route should be lowered to 50.

This table lists the goals and the community values.

Goal	Community Value
Set local preference of 50.	100:17
Set local preference of 150.	100:18
Prepend the AS path once when sending the network to external neighbors.	100:21
Prepend the AS path twice when sending the network to external neighbors.	100:22
Prepend the AS path three times when sending the network to external neighbors.	100:23

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.

Configuring BGP Communities

Several activities are required to successfully deploy [BGP](#) communities in a BGP-based network.

Configure BGP communities as follows:

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

To configure BGP communities on customers side, you first have to configure a route map in which you will set the community. Also the community propagation to BGP neighbor has to be configured, otherwise the community values attached to outgoing BGP updates will be striped out.

On the other side, service provider has to define BGP community lists to match routes based on attached BGP communities. Route maps have to be used to filter or modify BGP routing updates based on matching on community lists in incoming or outgoing updates.

1. Configuring Route Tagging with BGP Communities

First step is to set communities, which requires a route map.

```
router(config)#  
route-map name  
  match condition  
  set community value [ value ... ] [additive]
```

- Route tagging with communities is always done with a route map.
- Communities that are specified in the **set** keyword overwrite existing communities unless you specify the **additive** option.

```
router(config-router)# neighbor ip-address route-map map in | out
```

- You have to apply a route map to inbound or outbound BGP updates.

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

- You can apply a route map to redistributed routes

In a route-map configuration mode, you should use the **set community** command to attach a community attribute (or a set of communities) to a route. You can attach up to 32 communities to a single route with one route map set statement. If you use the keyword **additive**, 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 a route map to incoming or outgoing updates. You can also use it with redistribution from another routing protocol.



A route map is a filtering mechanism that has an "implicit deny" for all networks that are not matched in any route map statement. If a route map is not intended to filter routes, then you should add another route map statement at the end to permit all remaining networks without changing it (no **match** and no **set** commands are used within that route map statement).

Originally, Cisco IOS software accepted and displayed BGP community values as a single 32-bit value in a digital format. Newer Cisco IOS versions support the new format, where you can set or view a community as two colon-separated 16-bit numbers. The global command **ip bgp-community new-format** is recommended on all routers whenever communities contain the [AS](#) number.

2. Configuring Community Propagation

Second step is to enabling community propagation per neighbor for all internal neighbors. If communities are sent to external neighbors, you must enable community propagation for external neighbors.

```
router(config-router)# neighbor ip-address send-community
```

- By default, communities are stripped in outgoing BGP updates.
- You must manually configure community propagation to BGP neighbors.
- BGP peer groups are ideal for configuring BGP community propagation toward many neighbors.

A command that network administrators commonly forget when configuring BGP communities is the **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. You can also apply this command to a peer group.

3. Defining BGP Community Lists

On a service provider side, community lists have to be created to be used within route maps to match on community values.

```
router(config)# ip community-list 1-99 permit|deny value [ value ... ]
```

- This command defines a simple community list.
- You can use the keyword **internet** to match any community.

```
router(config)# ip community-list 100-199 permit|deny regexp
```

- This command defines an expanded community list.
- Communities that are attached to a route are ordered, converted to string, and matched with regular expression.
- Use "." to match any community value.

You can use a **standard community 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.

Comparison of Community Lists to Standard IP Access Lists

Similarities	Differences
The router evaluates the lines in the community-list sequentially.	The keyword internet should be used to permit any community value.
If no line matches communities that are attached to a BGP route, the route is implicitly denied.	If more values are listed in a single line, they all have to be in an update to produce a match.

An **expanded 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. This example shows how the process is accomplished:

1. The original list of communities in an update:
 - "10.0.0.0/24, NH=1.1.1.1, origin=I, AS-path=20 30 40, community=10:101, community=10:201, community=10:105, community=10:205"
2. A string of characters containing an ordered list of community values:
 - "_10:101_10:105_10:201_10_205_" ("_" represents a space)
3. A regular expression:
 - "permit _10:.0[1-5]_" ("_" represents an underscore that matches spaces)
4. The result:
 - This regular expression permits the route because it permits all routes with communities where the first 16 bits carry the AS number 10 and the second

- 16 bits contain 0 as the second digit and a number between 1 and 5 as the third digit; the first digit can be anything (as indicated by the ".").
5. Use regular expression ".*" to permit any community.

4. Matching BGP Communities with Route Maps

On a service provider side, route maps where community lists are used to match on community values have to be created. You can then use route maps to filter based on community values or to set other parameters or attributes (for example, local preference, [MED](#), or AS path prepending).

These route maps are then applied to incoming or outgoing updates.

```
router(config)#
route-map name permit | deny
  match community clist-number [exact]
  set attributes
```

- Community lists are used in match conditions in route maps to match on communities that are attached to BGP routes.
- **Route selection:**
 - You can use route maps to set weights, local preference, or metric based on BGP communities that are attached to the BGP route.
 - Normal route selection rules apply afterward. Routes not accepted by route map are dropped.
- **Default filters:**
 - Routes that are tagged with community **no-export** are sent to IBGP peers and intra-confederation EBGP peers.
 - Routes that are tagged with **local-as** are sent to IBGP peers.
 - Routes that are tagged with **no-advertise** are not sent in any outgoing BGP updates.

Use route maps to match the networks that carry a subset of communities that the community list permits. Other parameters or attributes can then be set based on community values. A route map with a community list matches a route if at least some communities that are attached to the route match the community list. 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 a 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).

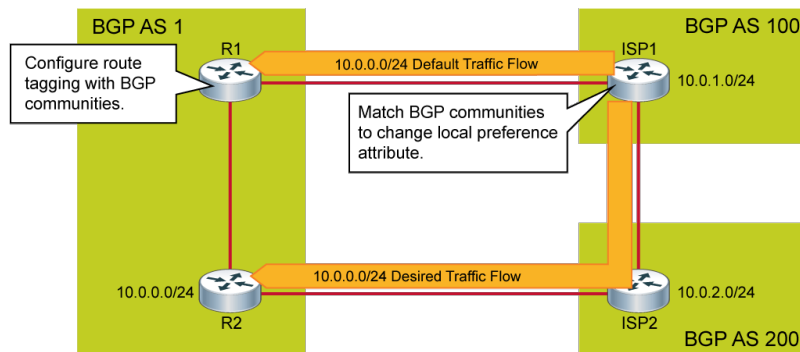
As mentioned before, there are some predefined community values that cause routers to automatically filter routing updates:

- **no-advertise:** If a router receives an update carrying this community, the router will not forward it to any neighbor.
- **local-as:** This community has a similar meaning to **no-export**, but it keeps a route within the local subautonomous system. The route is not propagated to intra-confederation external neighbors or to any other external neighbors.
- **no-export:** If a router receives an update carrying this community, the router will not propagate it to any external neighbors except to intra-confederation external neighbors.
- **internet:** This value advertises this route to the Internet community, to which all routers belong.

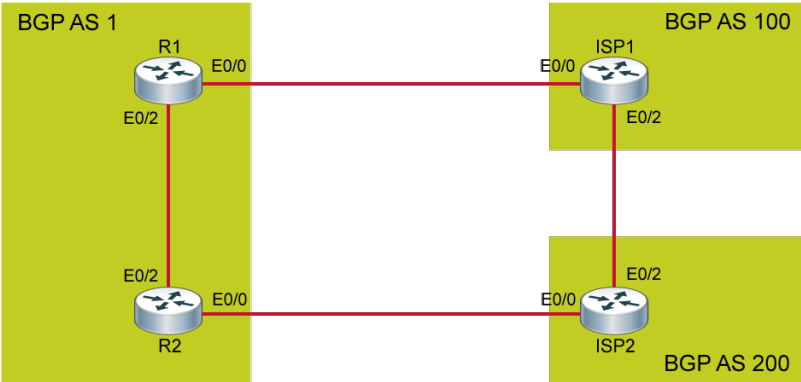
Discovery 16: Configure Local Preference Using the Communities

Overview

In this discovery, you will influence the returning path. You will configure local preference through the communities. You will do the configuration on the customer and on the service provider side.



Topology



Job Aids



If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

Device Information

Device	Interface	IP address	Description
R1	Ethernet 0/0	172.16.11.1/24	Connection to ISP1
R1	Ethernet 0/2	192.168.12.1/24	Connection to R2
R2	Ethernet 0/0	172.16.22.2/24	Connection to ISP2
R2	Ethernet 0/2	192.168.12.2/24	Connection to R1
R2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5	10.0.0.1/28 10.0.0.17/28 10.0.0.33/28 10.0.0.49/28 10.0.0.65/28	Loopbacks simulate LAN networks
ISP1	Ethernet 0/0	172.16.11.11/24	Connection to R1
ISP1	Ethernet 0/2	172.16.100.11/24	Connection to ISP2
ISP1	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6	10.0.1.1/28 10.0.1.17/28 10.0.1.33/28 10.0.1.49/28 10.0.1.65/28 10.0.1.81/28	Loopbacks simulate LAN networks
ISP2	Ethernet 0/0	172.16.22.22/24	Connection to R2
ISP2	Ethernet 0/2	172.16.100.22/24	Connection to ISP1
ISP2	Loopback 1 Loopback 2 Loopback 3 Loopback 4 Loopback 5 Loopback 6 Loopback 7 Loopback 8 Loopback 9	10.0.2.1/28 10.0.2.17/28 10.0.2.33/28 10.0.2.49/28 10.0.2.65/28 10.0.2.81/28 10.0.2.97/28 10.0.2.113/28 10.0.2.129/28	Loopbacks simulate LAN networks

[IP addresses](#) and advertised networks in [BGP](#) are preconfigured as shown below:



BGP is also preconfigured:

- [EBGP](#)
 - R1 to ISP1
 - R2 to ISP2
 - ISP1 to ISP2

- [IBGP](#)
 - R1 to R2

Configure Local Preference Using the Communities

Discovery Steps

Step 1

On the ISP1 router, verify initial best path to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 31, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network        Next Hop        Metric LocPrf Weight Path
*    10.0.0.0/28    172.16.100.22          0 200 1 i
*>    10.0.0.16/28    172.16.100.22          0 200 1 i
*>    10.0.0.32/28    172.16.100.22          0 200 1 i
*>    10.0.0.48/28    172.16.100.22          0 200 1 i
*>    10.0.0.64/28    172.16.100.22          0 200 1 i
*>    10.0.1.0/28     0.0.0.0                0 32768 i
*>    10.0.1.16/28    0.0.0.0                0 32768 i
*>    10.0.1.32/28    0.0.0.0                0 32768 i
*>    10.0.1.48/28    0.0.0.0                0 32768 i
   Network        Next Hop        Metric LocPrf Weight Path
*>    10.0.1.64/28    0.0.0.0                0 32768 i
*>    10.0.1.80/28    0.0.0.0                0 32768 i
*    10.0.2.0/28     172.16.11.1            0 1 200 i
*>    10.0.2.16/28    172.16.100.22          0 200 i
*    10.0.2.32/28    172.16.11.1            0 1 200 i
*>    10.0.2.48/28    172.16.100.22          0 200 i
*    10.0.2.64/28    172.16.11.1            0 1 200 i
*>    10.0.2.80/28    172.16.100.22          0 200 i
*    10.0.2.96/28    172.16.11.1            0 1 200 i
*>    10.0.2.112/28   172.16.100.22          0 200 i
*    10.0.2.128/28   172.16.11.1            0 1 200 i
*>    10.0.2.144/28   172.16.100.22          0 200 i
```

ISP1 received two routes to 10.0.0.0/24 networks:

- First route is received from ISP2 (next hop 172.16.100.22)
- Second route is received from R1 (next hop 172.16.11.1),

Since weight and local preference attributes are left on default values, ISP1 chooses the path with shortest [AS](#) path length as the best (the route that is received from R1).

If you want to influence the returning traffic to AS 1 from AS 100 to go via AS 200 (and not directly), you can do it by using community lists in AS 1. In the update from that AS, you can send a community list. This community list can then influence the value of, for example, local preference in AS 100 for all routes that are received from AS 1.

Step 2

Configure community list to influence the returning traffic path to your AS (AS 1). Use community for changing the local preference value on ISP's side. First do the customer's side of the configuration and configure route map on the R1 router to tag routes with BGP communities.

Name the route map "SetCom." Select all BGP routes and set community value of "100:17." You have already agreed with a service provider, that "100:17" community will set local preference of 50 on a service provider's side. Since the default local preference is 100, lower local preference (50) means that the route will not be chosen as the best one.

```
R1(config)# route-map SetCom permit 10
R1(config-route-map)# set community 100:17
```

Because you did not enter any **match** statement, all networks are permitted. If you wanted to set communities on specific routes, you could use a standard access list to match against, with the **match ip address** command in the route map.

Step 3

On the R1 router, apply the "SetCom" route map to all outgoing updates to the ISP1 EBGp neighbor.

```
R1(config)# router bgp 1
R1(config-router)# neighbor 172.16.11.11 route-map SetCom out
```

Step 4

Remember, that you have to enable propagation of community attributes to BGP neighbors. Otherwise the router will strip out any community values attached to outgoing updates.

Enable community propagation on the R1 router.

```
R1(config)# router bgp 1
R1(config-router)# neighbor 172.16.11.11 send-community
```

You enabled community propagation from AS 1 to AS 100.

Step 5

Now move on the service provider's side and define BGP community list that will match communities set in AS 1.

On the ISP1 router, configure community list to match communities that were previously set on the R1 router.

```
ISP1(config)# ip community-list 7 permit 100:17
```

Step 6

The last thing that you have to do is to translate received communities from AS 1 into local preference. On the ISP1 router, configure route map to match BGP communities and to change the local preference value of matched communities.

On the ISP1 router, configure route map "SetLocalPref," that will use the configured community list to find a community 100:17 and will set local-preference of 50. Do not forget to permit all other routes, that do not contain the right community, without changing anything in the update.

```
ISP1(config)# route-map SetLocalPref permit 10
ISP1(config-route-map)# match community 7
ISP1(config-route-map)# set local-preference 50
ISP1(config-route-map)# exit
ISP1(config)# route-map SetLocalPref permit 1000
```

Route map uses a community list, that was configured in previous step, to find community 100:17. If the community list matches one of the community attributes, the **set** command is executed, the local-preference is changed, and the route is permitted.

If the route does not contain the right community, the route is simply permitted by route map statement 1000 without changing anything in the update.

Step 7

On the ISP1 router, apply the configured route map to all incoming updates from R1 EBGp neighbor.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.11.1 route-map SetLocalPref in
```

The result will be that AS 100 will set local preference of 50 to all routes received from AS 1. So, it will prefer the other path (the path via AS 200) to reach AS 1.

Step 8

On the ISP1 router, re-establish BGP session to enforce the application of new route map.

To reestablish BGP session, use the **clear ip bgp *** command.

```
ISP1# clear ip bgp *
```

Step 9

On the ISP1 router, verify new received routes to reach the 10.0.0.0/24 networks in AS 1.

```
ISP1# show ip bgp
BGP table version is 21, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.0.0/28	172.16.100.22			0	200 1 i
*	172.16.11.1		50	0	1 i
*> 10.0.0.16/28	172.16.100.22			0	200 1 i
*	172.16.11.1		50	0	1 i
*> 10.0.0.32/28	172.16.100.22			0	200 1 i
*	172.16.11.1		50	0	1 i
*> 10.0.0.48/28	172.16.100.22			0	200 1 i
*	172.16.11.1		50	0	1 i
*> 10.0.0.64/28	172.16.100.22			0	200 1 i
*	172.16.11.1		50	0	1 i
*> 10.0.1.0/28	0.0.0.0	0		32768	i
*> 10.0.1.16/28	0.0.0.0	0		32768	i
*> 10.0.1.32/28	0.0.0.0	0		32768	i
*> 10.0.1.48/28	0.0.0.0	0		32768	i
Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.1.64/28	0.0.0.0	0		32768	i
*> 10.0.1.80/28	0.0.0.0	0		32768	i
*> 10.0.2.0/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.16/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.32/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.48/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.64/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.80/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.96/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.112/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i
*> 10.0.2.128/28	172.16.100.22	0		0	200 i
*	172.16.11.1		50	0	1 200 i

ISP1 again received two routes to 10.0.0.0/24 networks:

- The first route is received from ISP2 (next hop 172.16.100.22), with default local preference of 100. (Remember that the applied default local preference is not displayed.)
- The second route is received from R1 (next hop 172.16.11.1), with changed local preference of 50.

ISP1 chooses the path that is received from ISP2 as the best based on the highest local-preference value.

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

- Communities are displayed in **show ip 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 that match a community list can be displayed.
- All routes in a BGP table that have at least one attached community are displayed in **show ip bgp community** printout.

If the keyword **community** is included in **show ip bgp** command, all networks that have at least one community attribute are displayed. However, you can view communities only if you use the **show ip bgp prefix** command.

If you use the **show ip bgp community-list clist** command, all networks that the community list permits are listed. Using the **show ip bgp community as:nn [as:nn ...]** command, all routes in a BGP table having all the specified communities that are attached are displayed. If the keyword **exact** is added at the end, only the networks that match exactly are displayed.

Step 10

On the ISP1 router, verify all the networks, that have at least one community attribute.

```
ISP1# show ip bgp community
BGP table version is 21, local router ID is 10.0.1.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

   Network          Next Hop           Metric LocPrf Weight Path
*   10.0.0.0/28      172.16.11.1           50         0 1 i
*   10.0.0.16/28     172.16.11.1           50         0 1 i
*   10.0.0.32/28     172.16.11.1           50         0 1 i
*   10.0.0.48/28     172.16.11.1           50         0 1 i
*   10.0.0.64/28     172.16.11.1           50         0 1 i
*   10.0.2.0/28      172.16.11.1           50         0 1 200 i
*   10.0.2.16/28     172.16.11.1           50         0 1 200 i
*   10.0.2.32/28     172.16.11.1           50         0 1 200 i
*   10.0.2.48/28     172.16.11.1           50         0 1 200 i
*   10.0.2.64/28     172.16.11.1           50         0 1 200 i
*   10.0.2.80/28     172.16.11.1           50         0 1 200 i
*   10.0.2.96/28     172.16.11.1           50         0 1 200 i
*   10.0.2.112/28    172.16.11.1           50         0 1 200 i
*   10.0.2.128/28    172.16.11.1           50         0 1 200 i
```

You can see that all networks, 10.0.0.0/24 and 10.0.2.0/24, received from AS 1 are listed in the output. They are listed because you configured route tagging with BGP communities on R1 in AS 1, for all updates that are sent to AS 100.

Step 11

On the ISP1 router, verify the community attribute for one of the 10.0.0.0/24 networks, the 10.0.0.0/28 network.

```
ISP1# show ip bgp 10.0.0.0/28
BGP routing table entry for 10.0.0.0/28, version 2
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
    3
  Refresh Epoch 2
  200 1
    172.16.100.22 from 172.16.100.22 (10.0.2.129)
      Origin IGP, localpref 100, valid, external, best
  Refresh Epoch 1
  1
    172.16.11.1 from 172.16.11.1 (172.16.11.1)
      Origin IGP, localpref 50, valid, external
      Community: 6553617
```

You can see that ISP1 received two routes to 10.0.0.0/28 network:

- First route is received from ISP2, with default local preference of 100.
- Second route is received from R1, with the community attribute 6553617 and consequently it has changed local preference of 50.

Note that the community is displayed as a single 32-bit value (6553617). If you want it to be in a form of *as:nn*, you have to configure the support for new-format on the router, using the **ip bgp-community new-format** command.

Step 12

On the ISP1 router, enable new BGP community format.

```
ISP1(config)# ip bgp-community new-format
```

Step 13

On the ISP1 router, again verify the community attribute for 10.0.0.0/28 network.

```
ISP1# show ip bgp 10.0.0.0/28
BGP routing table entry for 10.0.0.0/28, version 2
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
    3
  Refresh Epoch 2
  200 1
    172.16.100.22 from 172.16.100.22 (10.0.2.129)
      Origin IGP, localpref 100, valid, external, best
  Refresh Epoch 1
  1
    172.16.11.1 from 172.16.11.1 (172.16.11.1)
      Origin IGP, localpref 50, valid, external
      Community: 100:17
```

Now the community is displayed as two colon-separated 16-bit numbers, 100:17.

BGP Named Community Lists

The [BGP](#) Named Community Lists feature introduces a new type of community list that is called the named community list. A named community list can be configured with regular expressions or with numbered community lists.

- Allows you to assign meaningful names to community lists and increases the number of community lists that can be configured.
- Can be configured with regular expressions and with numbered community lists.
- Increases the number of community lists that a network operator can configure —there is no limitation on the number of named community lists that can be configured.

Commands:

- Use the **ip community-list** command to create a numbered or named community list.
- Use the **match community** command in route-map configuration mode, to match a BGP community.
- Use the **set com-list delete** command in route-map configuration mode to remove communities from the community attribute of an update.

The BGP Named Community Lists feature allows you to assign meaningful names to community lists. All rules of numbered communities apply to named community lists, with one exception. The exception is that there is no limitation on the number of community attributes that can be configured for a named community list. Both, standard and expanded, community lists 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.

ip community-list Command

To create a numbered or named community list for BGP and to control access to it, use the **ip community-list** command in global configuration mode. To delete the community list, use the **no** form of this command.

```
ip community-list {standard-list-number | expanded-list-number [regular-expression] | {standard | expanded} community-list-name} {permit | deny} community-number | regular-expression
```

Syntax Description

Parameter	Description
<i>standard-list-number</i>	Specifies a standard community list number from 1 to 99 that identifies one or more permit or deny groups of communities.
<i>expanded-list-number</i>	Specifies an expanded community list number from 100 to 199 that identifies one or more permit or deny groups of communities.
standard	Configures a standard named community list.
expanded	Configures an expanded named community list.
<i>community-list-name</i>	The community list name.
permit	Permits access for a matching condition.
deny	Denies access for a matching condition.
<i>community-number</i>	Community number that you configure with the set community command.

Valid value of community number is one of the following:

- A number from 1 to 4294967200. You can specify a single number or multiple numbers and you separate them by a space.
- **internet**—The Internet community.
- **no-export**—Routes with this community are sent to peers in other subautonomous systems within a confederation. Do not advertise this route to an [EBGP](#) peer. External systems are the systems outside the confederation. If there is no confederation, an external system is any EBGP peer.
- **local-as**—Send this route to peers in other subautonomous systems within the local confederation. Do not advertise this route to an external system.

- **no-advertise**—Do not advertise this route to any peer (internal or external).

match community Command

To match a BGP community, use the **match community** command in route-map configuration mode.

To remove the **match community** command from the configuration file and restore the system to its default condition where the software removes the BGP community list entry, use the **no** form of this command.

```
match community standard-list-number | expanded-list-number | community-list-name [exact]
```

Syntax Description

Parameter	Description
<i>standard-list-number</i>	Specifies a standard community list number from 1 to 99 that identifies one or more permit or deny groups of communities.
<i>expanded-list-number</i>	Specifies an expanded community list number from 100 to 199 that identifies one or more permit or deny groups of communities.
<i>community-list-name</i>	The community list name.
exact	(Optional) Indicates that an exact match is required. All of the communities and only those communities that are specified must be present.

set comm-list delete Command

To remove communities from the community attribute of an inbound or outbound update, use the **set comm-list delete** command in route-map configuration mode.

To negate a previous **set comm-list delete** command, use the **no** form of this command.

```
set comm-list community-list-number | community-list-name delete
```

Syntax Description

Parameter	Description
<i>community-list-number</i>	A standard or expanded community list number.
<i>community-list-name</i>	A standard or expanded community list name.

BGP Named Community List Examples

The following example creates a standard community list that permits all routes except the routes with the communities 5 and 10 or 10 and 15:

```
Router(config)# ip community-list 1 deny 5 10
Router(config)# ip community-list 1 deny 10 15
Router(config)# ip community-list 1 permit internet
```

The following example creates a standard community list that permits all routes within the local [AS](#):

```
Router(config)# ip community-list 1 permit local-as
```

The following example creates a standard named community list with the name COMMUNITY_A that permits all routes within the local AS and denies all routes with the internet community attribute:

```
Router(config)# ip community-list standard COMMUNITY_A permit local-AS
Router(config)# ip community-list standard COMMUNITY_A deny internet
```

The following example creates an expanded named community list with the name COMMUNITY_B that will not advertise routes to EBGp peers:

```
Router(config)# ip community-list expanded COMMUNITY_B permit no-export
```

The following example creates a named community list with the name COMMUNITY_C that will not advertise this route to any EBGp or EBGp peers:

```
Router(config)# ip community-list expanded COMMUNITY_C permit no-advertise
```

The following example uses a regular expression. The example creates a filter that will deny all communities that contain a number:

```
Router(config)# ip community-list 100 deny [0-9]*
```

BGP Cost Community

The cost community is a nontransitive extended community attribute that is passed to [IBGP](#) and confederation peers but not to [EBGP](#) peers.

- Allows you to customize the BGP best-path selection process for a local AS or confederation.
- Applied to internal routes by configuring the **set extcommunity cost** command in a route map.
- Influences the BGP best-path selection process at the POI.
- Can be used as a "tie breaker" during the best-path selection process.

The configuration of the [BGP](#) Cost Community feature allows you to customize the BGP best-path selection process for a local [AS](#) or confederation by assigning cost values to specific routes.

To create a set clause to apply the cost community attribute to routes that pass through a route map, use the **set extcommunity cost** command in route-map configuration mode. To delete the cost community set clause, use the **no** form of this command.

```
set extcommunity cost[igp] community-id cost-value
```

Syntax Description

Parameter	Description
igp	The IGP POI . The configuration of this keyword forces the cost community to be evaluated after the IGP distance to the next hop has been compared.
<i>community-id</i>	The ID for the configured extended community. The range is from 0 to 255.
<i>cost-value</i>	The configured cost that is set for matching paths in the route map. The range is from 0 to 4294967295.

The cost community set clause is configured with a cost community ID number (0 to 255) and cost number (0 to 4294967295). The cost number value determines the preference for the path. The path with the lowest cost community number is preferred. Paths that are not specifically configured with the cost community attribute are assigned a default cost number value of 2147483647 (the midpoint between 0 and 4294967295) and evaluated by the best-path selection process accordingly. When two paths have been configured with the same cost number value, the path selection process prefers the path with the lowest cost community ID. The cost extended community attribute is propagated to IBGP peers when extended community exchange is enabled with the **neighbor send-community** command.

The cost community attribute influences the BGP best-path selection process at the POI. By default, the POI follows the IGP metric comparison. When BGP receives multiple paths to the same destination, it uses the best-path selection process to determine which path is the best path. BGP automatically makes the decision and installs the best path into the routing table. The POI allows you to assign a preference to a specific path when multiple equal-cost paths are available. If the POI is not valid for local best-path selection, the cost community attribute is silently ignored.

Multiple paths can be configured with the cost community attribute for the same POI. The path with the lowest cost community ID is considered first. In other words, all of the cost community paths for a specific POI are considered, starting with the one with the lowest cost community ID. Paths that do not contain the cost community (for the POI and community ID being evaluated) are assigned the default community cost value (2147483647). If the cost community values are equal, then cost community comparison proceeds to the next-lowest community ID for this POI.

Applying the cost community attribute at the POI allows you to assign a value to a path originated or learned by a peer in any part of the local AS or confederation. The cost community can be used as a "tie breaker" during the best path selection process. Multiple instances of the cost community can be configured for separate equal-cost paths within the same AS or confederation. For example, a lower-cost community value can be applied to a specific exit path in a network with multiple

equal-cost exit points. Then, the BGP best-path selection process will prefer the specific exit path.

BGP Cost Community Example

The following example configuration shows the configuration of the **set extcommunity cost** command. This example applies the cost community ID of 1 and cost community value of 100 to the routes that the route map permits. This configuration will cause the best-path selection process to prefer this route over other equal-cost paths that this route map sequence does not permit.

```
Router(config)# router bgp 50000
Router(config-router)# neighbor 10.0.0.1 remote-as 50000
Router(config-router)# neighbor 10.0.0.1 update-source Loopback 0
Router(config-router)# address-family ipv4
Router(config-router-af)# neighbor 10.0.0.1 activate
Router(config-router-af)# neighbor 10.0.0.1 route-map COST1 in
Router(config-router-af)# neighbor 10.0.0.1 send-community both
Router(config-router-af)# exit
Router(config)# route-map COST1 permit 10
Router(config-route-map)# match ip-address 1
Router(config-route-map)# set extcommunity cost 1 100
```

BGP Link Bandwidth Feature

The [BGP](#) Link Bandwidth feature is used to enable multipath load balancing for external links with unequal bandwidth capacity. This feature supports [IBGP](#), [EBGP](#) multipath load balancing, and EIGRP multipath load balancing in [MPLS VPNs](#). When this feature is enabled, routes learned from directly connected external neighbors are propagated through the IBGP network with the bandwidth of the source external link.

- Use it to enable multipath load balancing for external links with unequal bandwidth capacity.
- You enable it under an IPv4 or VPNv4 address family sessions by entering the **bgp dmzlink-bw** command.
- Routes that are learned from directly connected external neighbor are propagated through the IBGP network with the bandwidth of the source external link.

You enable the BGP Link Bandwidth feature under an [IPv4](#) or [VPNv4](#) address-family session by entering the **bgp dmzlink-bw** command.

The BGP Link Bandwidth feature allows BGP to be configured to send traffic over multiple IBGP or EBGP learned paths. The traffic that is sent on those paths is proportional to the bandwidth of the links that are used to exit the [AS](#). The configuration of this feature can be used with EBGP and IBGP multipath features to enable unequal-cost load balancing over multiple links. Unequal-cost load balancing over links with unequal bandwidth was not possible in BGP before the BGP Link Bandwidth feature was introduced.

The link bandwidth extended community attribute indicates the preference of an AS exit link in terms of bandwidth. This extended community is applied to external links between directly connected EBGP peers by entering the **neighbor dmzlink-bw** command. The link bandwidth extended community attribute is propagated to IBGP peers when extended community exchange is enabled with the **neighbor send-community** command.

bgp dmzlink-bw Command

To configure BGP to distribute traffic proportionally over external links with unequal bandwidth when multipath load balancing is enabled, use the **bgp dmzlink-bw** command in address family configuration mode. To disable traffic distribution proportional to the link bandwidth, use the **no** form of this command.

```
bgp dmzlink-bw
```



This command has no keywords or arguments.

neighbor dmzlink-bw Command

To configure BGP to advertise the bandwidth of links that are used to exit an AS, use the **neighbor dmzlink-bw** command in address family configuration mode. To disable link bandwidth advertisement, use the **no** form of this command.

```
neighbor ip-address dmzlink-bw
```

Syntax Description

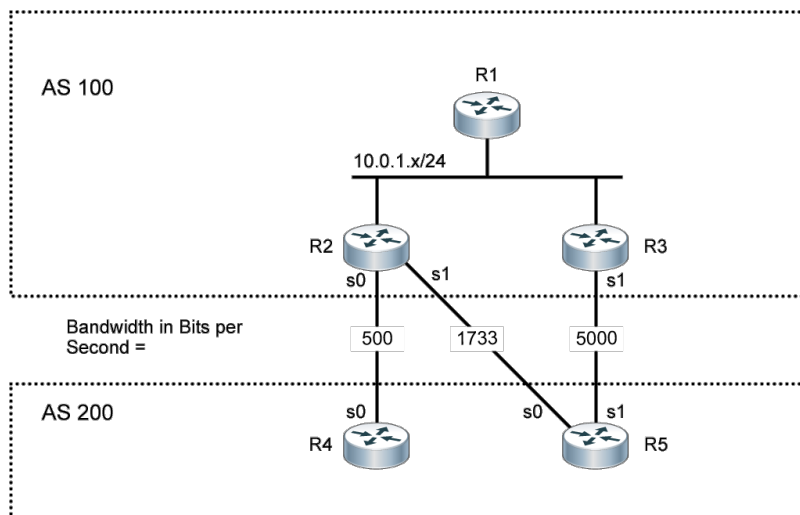
Parameter	Description
<i>ip-address</i>	The IP address that identifies the external interface

The link bandwidth extended community attribute is a 4-byte value that is configured for a link on the [DMZ](#) interface that connects two single-hop EBGP peers. The link bandwidth extended community attribute is used as a traffic-sharing value relative to other paths while forwarding traffic. Two paths are designated as equal for load balancing if the weight, local preference, AS path length, [MED](#), and [IGP](#) costs are the same.

BGP Link Bandwidth Example

In this example, the BGP Link Bandwidth feature is configured so that BGP will

distribute traffic proportionally to the bandwidth of each external link.



The figure shows two external autonomous systems. Three links connect the two ASs, and each AS carries a different amount of bandwidth (unequal-cost links). Multipath load balancing is enabled, and traffic is balanced proportionally.

First R1 is configured to support IBGP multipath load balancing and to exchange the BGP extended community attribute with IBGP neighbors, R2 on 10.0.1.2 IP address and R3 on 10.0.1.3 IP address.

```
R1(config)# router bgp 100
R1(config-router)# neighbor 10.0.1.2 remote-as 100
R1(config-router)# neighbor 10.0.1.2 update-source Loopback 0
R1(config-router)# neighbor 10.0.1.3 remote-as 100
R1(config-router)# neighbor 10.0.1.3 update-source Loopback 0
R1(config-router)# address-family ipv4
R1(config-router-af)# bgp dmzlink-bw
R1(config-router-af)# neighbor 10.0.1.2 activate
R1(config-router-af)# neighbor 10.0.1.2 send-community both
R1(config-router-af)# neighbor 10.0.1.3 activate
R1(config-router-af)# neighbor 10.0.1.3 send-community both
R1(config-router-af)# maximum-paths ibgp 6
```

R2 is configured to support multipath load balancing, to distribute R4 (172.16.100.1 IP address) and R5 (172.16.200.2 IP address) link traffic proportionally to the bandwidth of each link, and to advertise the bandwidth of these links to IBGP neighbors as an extended community:

```
R2(config)# router bgp 100
R2(config-router)# neighbor 10.0.1.1 remote-as 100
R2(config-router)# neighbor 10.0.1.1 update-source Loopback 0
R2(config-router)# neighbor 10.0.1.3 remote-as 100
R2(config-router)# neighbor 10.0.1.3 update-source Loopback 0
R2(config-router)# neighbor 172.16.100.1 remote-as 200
R2(config-router)# neighbor 172.16.100.1 ebgp-multihop 1
R2(config-router)# neighbor 172.16.200.2 remote-as 200
R2(config-router)# neighbor 172.16.200.2 ebgp-multihop 1
R2(config-router)# address-family ipv4
R2(config-router-af)# bgp dmzlink-bw
R2(config-router-af)# neighbor 10.0.1.1 activate
R2(config-router-af)# neighbor 10.0.1.1 next-hop-self
R2(config-router-af)# neighbor 10.0.1.1 send-community both
R2(config-router-af)# neighbor 10.0.1.3 activate
R2(config-router-af)# neighbor 10.0.1.3 next-hop-self
R2(config-router-af)# neighbor 10.0.1.3 send-community both
R2(config-router-af)# neighbor 172.16.100.1 activate
R2(config-router-af)# neighbor 172.16.100.1 dmzlink-bw
R2(config-router-af)# neighbor 172.16.200.2 activate
R2(config-router-af)# neighbor 172.16.200.2 dmzlink-bw
R2(config-router-af)# maximum-paths ibgp 6
R2(config-router-af)# maximum-paths 6
```

Also R3 is configured to support multipath load balancing and to advertise the bandwidth of the link with R5 (172.16.300.3 IP address) to IBGP neighbors as an extended community:

```
R3(config)# router bgp 100
R3(config-router)# neighbor 10.0.1.1 remote-as 100
R3(config-router)# neighbor 10.0.1.1 update-source Loopback 0
R3(config-router)# neighbor 10.0.1.2 remote-as 100
R3(config-router)# neighbor 10.0.1.2 update-source Loopback 0
R3(config-router)# neighbor 172.16.300.3 remote-as 200
R3(config-router)# neighbor 172.16.300.3 ebgp-multihop 1
R3(config-router)# address-family ipv4
R3(config-router-af)# bgp dmzlink-bw
R3(config-router-af)# neighbor 10.0.1.1 activate
R3(config-router-af)# neighbor 10.0.1.1 send-community both
R3(config-router-af)# neighbor 10.0.1.1 next-hop-self
R3(config-router-af)# neighbor 10.0.1.2 activate
R3(config-router-af)# neighbor 10.0.1.2 send-community both
R3(config-router-af)# neighbor 10.0.1.2 next-hop-self
R3(config-router-af)# neighbor 172.16.300.3 activate
R3(config-router-af)# neighbor 172.16.300.3 dmzlink-bw
R3(config-router-af)# maximum-paths ibgp 6
R3(config-router-af)# maximum-paths 6
```

BGP Support for Sequenced Entries in Extended Community Lists

The [BGP](#) Sequenced Entries in Extended Community Lists feature allows you automatic sequencing of individual entries in BGP extended community lists. It also provides you the ability to remove or resequence extended community list entries without deleting the entire existing extended community list.

- Allows you automatic sequencing of individual entries in BGP extended community lists.
- Provides you 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 existing sequence numbers for extended community list entries.
- Configures an extended community list to use default values.

Both named and numbered extended community lists can be configured in IP extended community list configuration mode. To enter IP extended community list configuration mode, issue the **ip extcommunity-list** command with either the **expanded** or **standard** keyword followed by the extended community list name. This configuration mode supports all of the functions that are available in global configuration mode. In addition, you can perform the following operations:

- Configure sequence numbers for extended community list entries.
- Resequences existing sequence numbers for extended community list entries.
- Configure an extended community list to use default values.

ip extcommunity-list Command

To create an extended community list and control access to it, use the **ip extcommunity-list** command in global configuration mode. To delete the entire community list, use the **no** form of this command.

```
ip extcommunity-list expanded-list-number | expanded list-  
name {permit| deny} [regular-expression] | standard-list-  
number | standard list-name {permit| deny} [rt extcom-value] [soo extcom-  
value]
```

Syntax Description

Parameter	Description
<i>expanded-list-number</i>	An expanded list number from 100 to 500 that identifies one or more permit or deny groups of extended communities.
<i>standard-list-number</i>	A standard list number from 1 to 99 that identifies one or more permit or deny groups of extended communities.
expanded <i>list-name</i>	Creates an expanded named extended community list and enters IP extended community list configuration mode.
standard <i>list-name</i>	Creates a standard named extended community list and enters IP extended community list configuration mode.
permit	Permits access for a matching condition.
deny	Denies access for a matching condition.
<i>regular-expression</i>	(Optional) An input string pattern to match against.
rt	(Optional) Specifies the RT extended community attribute. The rt keyword can be configured only with standard extended community lists and not expanded community lists.
soo	(Optional) Specifies the SOO extended community attribute. The soo keyword can be configured only with standard extended community lists and not expanded community lists.
<i>extcom-value</i>	Specifies the RT or SOO extended community value. The value can be one of the following combinations: <i>autonomous-system-number : network-number</i> or <i>ip-address : network-number</i>
<i>sequence-number</i>	(Optional) The sequence number of a named or numbered extended community list. This value can be a number from 1 to 2147483647.

Parameter	Description
	Community list sets a keyword or argument to default behavior or value.
exit	(Optional) Exits IP extended community list configuration mode.
resequence	(Optional) Changes the sequences of extended community list entries to the default sequence numbering or to the specified sequence numbering. Extended community entries are sequenced by 10-number increments by default.
<i>starting-sequence</i>	(Optional) Specifies the number for the first entry in an extended community list.
<i>sequence-increment</i>	(Optional) Specifies the increment range for each subsequent extended community list entry.

Sequenced and Resequenced Extended Community List Entry Example

The following example creates and configures a named extended community list that will permit routes only from RT 64512:10, 65000:20, 64535:30 and SOO 65535:40. All other routes are implicitly denied.

```
Router(config)# ip extcommunity-list standard NAMED_LIST
Router(config-extcom-list)# 1 permit rt 64512:10
Router(config-extcom-list)# 2 permit rt 65000:20
Router(config-extcom-list)# 3 permit rt 64535:30
Router(config-extcom-list)# 4 permit soo 65535:40
Router(config-extcom-list)# end
```

Resequence the extended community list entries in the named community list that is configured above. The first entry is resequenced to the number 50 and the range for each subsequent entry to follow by 100 (for example, 150, 250, 350, and so on).

```
Router(config)# ip extcommunity-list standard NAMED_LIST
Router(config-extcom-list)# resequence 50 100
Router(config-extcom-list)# end
```

To display routes that the named extended community list permits, use the **show ip extcommunity-list EXEC** command. The output shows the configuration from the first example after it has been resequenced with user-defined values.

```
Router> show ip extcommunity-list
Standard extended community-list NAMED_LIST
  50 permit RT:64512:10
 150 permit RT:65000:20
 250 permit RT:64535:30
 350 permit SoO:65535:40
```

Summary

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

- You can use the BGP community attribute to create an AS-wide routing policy or to provide services to neighboring autonomous systems.
- A community is an attribute that is used to tag BGP routes that you can use to manipulate path selection and enforce administrative policies.
- To configure community attribute, you should:
 - Configure a route map with a **set community** command.
 - Configure propagation of BGP communities on the routers on a per-neighbor basis
 - Configure community lists to match against the community attribute as a method of route selection
 - Configure a route map to match networks that carry communities that the community list permits and apply policies to them.
- The **BGP Named Community Lists** feature allows you to assign meaningful names to community lists and increases the number of community lists that can be configured.
- The configuration of the **BGP Cost Community** feature allows you to assign cost values to specific routes. This way, you customize the BGP best path selection process for a local AS or confederation.
- The **BGP Link Bandwidth** feature allows you multipath load balancing for external links with unequal bandwidth capacity.
- BGP Support for **Sequenced Entries in Extended Community Lists** allows automatic sequencing of individual entries in BGP extended community lists. It also enables you to remove or resequence extended community list entries without deleting the entire existing extended community list.

Module Summary

Overview

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

The following BGP attributes influence the route selection:

- **Weight** (The first criterion in BGP route selection.)
 - You can configure a default weight or use route-maps to configure it.
- **Local preference** (The second-strongest criterion in the route selection process.)
 - You can configure a default local preference or use route-maps to configure it.
- **AS-path**
 - You can perform AS-path prepending on outgoing EBGp updates over the undesired return path.
- **MED** (A "weak" parameter in the route selection process.)
- **Community**
 - Used to tag BGP routes that you can use to manipulate path selection and enforce administrative policies.

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. Number the 10 BGP selection criteria in the order in which they are used, from the first to the last, when selecting the BGP path that is submitted to the IP routing table. (Source: "Influencing BGP Route Selection with Weights")

Prefer the shortest AS path.	5
Prefer the path through the closest IGP neighbor.	4
Prefer the route that was originated by the local router.	6
Prefer an EBGp path over an IBGP path.	3
Prefer the path with the lowest neighbor BGP router ID.	2
Prefer the highest weight.	8
Prefer the oldest route for EBGp paths.	1
Prefer the lowest origin code (IGP < EGP < incomplete).	7
Prefer the lowest MED.	10
Prefer the highest local preference.	9

1. What is the default weight for routes that are received from a BGP neighbor? (Source: "Influencing BGP Route Selection with Weights")

0

100

32768

It depends on the Cisco IOS release.

1. When are the weights that are configured on a neighbor enforced?
(Source: "Influencing BGP Route Selection with Weights")

- Before the new weights can take effect, the BGP process on the router must be removed and reconfigured.

- The router must first be rebooted for the new weights to take effect.

- The new weights will be applied after the BGP update interval of 30 minutes expires.

- The new weight configuration is applied to all routes that are received following the configuration change.

1. If you want all routes received from the neighbor to have the new weight value applied, you have to reestablish BGP session. True or false? (Source: "Influencing BGP Route Selection with Weights")

☒ true

☐ false

1. How could you implement a primary/backup ISP routing policy by using weights? (Source: "Influencing BGP Route Selection with Weights")

- Assign higher weights to all routes that are received from the backup ISP.

- Assign lower weights to all routes that are received from the backup ISP.

- Assign higher weights to all routes that are received from the primary ISP.

- Assign lower weights to all routes that are received from the primary ISP.

1. When you are using route-maps to modify weights, what happens by default to a route that does not match any of the route-map statements? (Source: "Influencing BGP Route Selection with Weights")

☐ The route is accepted with the weight attribute unmodified.

☐ The route is discarded.

☐ The route is inserted into the BGP table but not into the IP routing table.

☐ An error is displayed on the router console and in router debugs.

1. Which method of influencing route selection with weights is the last to be applied on an incoming interface? (Source: "Influencing BGP Route Selection with Weights")

☐ prefix-list

☐ route-map

☐ filter-list weight

☐ default weight

1. You can set the weight using a route map applied to a neighbor in the outgoing direction. True or false? (Source: "Influencing BGP Route Selection with Weights")

☒ true

☐ false

1. Which two of the following statements are correct regarding BGP route selection? (Choose two.) (Source: "Influencing BGP Route Selection with Weights")

- ☐ If two routes have the same weight attribute, the route with the lowest local preference is chosen.
- ☐ The route with the highest weight is always chosen first.
- ☐ The weight attribute is global within an AS.
- ☐ The weight attribute is only local to the local router.
- ☐ The weight value is propagated by all routers.

1. What is a key difference between the local preference and weight attributes? (Source: "Setting BGP Local Preference")

- Local preference is local to the route on which it is configured.
- Local preference is local to the AS within which it has been configured.
- Local preference is local to the BGP administrative domain.
- Local preference is global to a BGP domain.

1. What is the default local preference for routes that are received from a BGP neighbor? (Source: "Setting BGP Local Preference")

0

100

There is no default local preference value.

It depends on the Cisco IOS release.

1. Which two statements regarding local preference are true? (Choose two.) (Source: "Setting BGP Local Preference")

- ☐ The higher value for local preference is preferred.
- ☐ Local preference is used only between EBGp neighbors
- ☐ The lower value for local preference is preferred.
- ☐ Local preference is used only between IBGP neighbors

1. Which two of the following statements about the influence of local preference on BGP route selection is accurate? (Choose two.)
(Source: "Setting BGP Local Preference")

- ☐ When you set local preference, you can view it on neighboring routers, but you must reset it.
- ☐ You can use local preference to ensure AS-wide route selection policy.
- ☐ Local preference is used to select routes with unequal weights.
- ☐ Local preference is the second-strongest criterion in the route selection process.

1. Let say we have two routes to the same destination, fist one has a weight of 50 and a local preference of 150, while the other one has a weight of 150 and a local preference of 50. The fist route will be selected as the best one, since it has higher local preference value. True or false? (Source: "Setting BGP Local Preference")

☐ true

☐ false

1. Which Cisco IOS command is used to change the default value of local preference? (Source: "Setting BGP Local Preference")

☐ **set local-preference**

☐ **bgp default local-preference**

☐ **show ip bgp**

☐ **show ip bgp *prefix***

1. Which Cisco IOS command is used to configure BGP local preference with route-map statements? (Source: "Setting BGP Local Preference")

☐ **set local-preference**

☐ **bgp default local-preference**

☐ **show ip bgp**

☐ **show ip bgp *prefix***

1. Locally applied default value can be displayed with **show ip bgp** command. True or false? (Source: "Setting BGP Local Preference")

☐ true

☐ false

1. Which Cisco IOS command is necessary to display the locally applied BGP value? (Source: "Setting BGP Local Preference")

☐ **show bgp preference detail**

☐ **show ip bgp**

☐ **show ip bgp detail**

☐ **show ip bgp *prefix***

1. Which two statements regarding the AS path are true? (Choose two.) (Source: "Using AS Path Prepending")

- ☐ The shorter AS path is preferred.
- ☐ The longer AS path is preferred.
- ☐ The AS path is prepended and exchanged between autonomous systems.
- ☐ The AS path is local to an AS.

1. What is AS path prepending? (Source: "Using AS Path Prepending")

- 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.

- 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.

- When a router, sending a BGP update, adds its AS number to the AS path attribute multiple times.

- When a router uses the AS path attribute in route selection.

1. The AS path will be the route selection criterion that is used when which of the following is true? (Source: "Using AS Path Prepending")

- ☐ It is the first criterion that is used in BGP route selection.
- ☐ It is used when there is no difference in weight, local preference, or route origination.
- ☐ It is used when the MED is identical on the candidate routes.
- ☐ The weight, local preference, MED, and origin attributes must be identical before the AS path attribute is used for route selection.

1. Which command do you use to manipulate the AS path attribute?
(Source: "Using AS Path Prepending")

☐ The global configuration command **set as-path prepend as-number**.

☐ The router configuration command **set as-path prepend as-number**.

☐ The **set as-path prepend as-number** command in a route-map.

☐ The interface global command **set as-path prepend as-number**.

1. The configuration below is from a router in AS 347, which is advertising network 11.0.0.0/8 to an EBGP neighbor 2.0.0.2 in AS 529. What are the contents of the AS path attribute for route 11.0.0.0/8 on a router that is residing in AS 529? (Source: "Using AS Path Prepending")
- ```
route-map addAS permit 10
set as-path prepend 347 347 347
router bgp 347
neighbor 2.0.0.2 remote-as 529
neighbor 2.0.0.2 route-map addAS out
```

347 347 347

347 347 347 347

529 347 347 347

529 347 347 347 347

1. Why do you need to use AS path prepending? (Source: "Using AS Path Prepending")

AS path prepending allows a customer to potentially influence return path route selection.

AS path prepending is used on a customer router to control outgoing route updates.

Service providers use AS path prepending to control incoming updates from a customer AS.

AS path prepending is used between service providers that are connected to the same customer AS to determine which will be the primary link to the customer.

1. Which regular expression will allow any AS path number, but with a length of one? (Source: "Using AS Path Prepending")

☐ `^[0-9]+$`

☐ `^[0-9](_[0-9])*$`

☐ `^_[0-9]$`

☐ `^[1-8]+$`



1. How does AS path prepending affect a router? (Source: "Using AS Path Prepending")

AS path prepending is simply a term that is used to describe when a router uses the AS path attribute in route selection and hence does not affect router resources.

The longer the AS path attribute attached to BGP updates, the more router memory requirements increase.

AS path prepending does not impact the router because Cisco IOS software recognizes that AS path prepending is in use and stores a single AS number with a pointer to the number of AS path prepends.

AS path prepending causes the router to operate in process-switching mode because the BGP update must be stored, manipulated, and then rewritten to accommodate the new AS path attribute.

1. Which two of the following are characteristics of the function of the BGP Hide Local-Autonomous System feature? (Choose two.)  
(Source: "Using AS Path Prepending")

- ☐ Allows you to transparently change the AS number for the entire BGP network.
- ☐ Ensures that routes can be propagated throughout the AS.
- ☐ Allows customization of the AS number for EBGP peer groupings through the **set as-path** command.
- ☐ Changes the AS number for all IBGP peers at the same time.

1. What is the typical application of the MED attribute? (Source: "Understanding BGP Multi-Exit Discriminators")

- To influence path selection out of an originating AS.
- To provide a strong metric to select the best path when multiple routes exist.
- To have a BGP attribute traversing many autonomous systems while influencing path selection.
- To influence the return path of traffic back into an AS.

1. What is the default MED value? (Source: "Understanding BGP Multi-Exit Discriminators")

0

100

There is no default MED value.

It depends on the Cisco IOS release.

1. What are three BGP attributes that are compared before the MED? (Choose three.) (Source: "Understanding BGP Multi-Exit Discriminators")

☐ largest weight

☐ originated routes

☐ AS path length

☐ lowest IP address

1. Which two statements about the Cisco IOS command that is required to configure changes to the default BGP MED on a Cisco IOS router are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")

- ☐ The MED is a mandatory attribute.
- ☐ Using the **default-metric** command in BGP configuration mode will cause one redistributed network to have the specified MED value.
- ☐ There is no MED attribute that is attached to a route by default.
- ☐ To set the default metric value (MED) for BGP routes, use the default-metric command.

1. Which two statements about the Cisco IOS commands that are required to configure changes to the BGP MED attribute with route-map statements are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")

☐ The **set metric** command is used within route-map configuration mode to set the MED attribute.

☐ The **neighbor address route-map name in | out** command applies a route-map to incoming updates from all neighbors.

☐ Per-neighbor MED is configured by using a route-map with a match condition.

☐ You can use a route-map to set the MED on incoming or outgoing updates.

1. Which two of the following statements about the Cisco IOS commands that are required to configure advanced MED features on Cisco routers are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")

▪ The **bgp bestpath med confed** command allows routers to compare paths learned from confederation peers.

▪ When you enable a deterministic MED comparison, you allow a router to compare MED values after it considers BGP route type (external or internal) and IGP metric to the next-hop address.

▪ Use the **bgp always-compare-med** command to force the router to compare the MED even if the paths come from different autonomous systems.

▪ Cisco IOS software, on the other hand, regards a missing MED attribute as having a value of 1.



1. If you configure inbound soft reconfiguration with a route-map and issue the **show ip bgp prefix** command, which value of the MED attribute is displayed? (Source: "Understanding BGP Multi-Exit Discriminators")

☐ Only the original route (no MED) is displayed.

☐ Both the original route and the modified route are displayed.

☐ Only the modified route is displayed.

☐ The MED attribute is not displayed with the **show ip bgp prefix** command.

1. Which of the following statements about the Cisco IOS commands that are required to troubleshoot BGP MED configurations on a Cisco router is accurate? (Source: "Understanding BGP Multi-Exit Discriminators")

To see the original MED, you need to enable hard reconfiguration on the router.

The command **show ip bgp neighbor *address* received-routes** displays the original updates before any filters or route-maps have filtered or changed them.

If hard reconfiguration is enabled, the original updates to the MED attribute are available by using the **show ip bgp *prefix*** command.

Issuing the **show ip route** command will display the MED value.

1. Which two statements regarding the MED are true? (Choose two.)  
(Source: "Understanding BGP Multi-Exit Discriminators")

- ☐ The higher value for the MED is preferred.
- ☐ The lower value for the MED is preferred.
- ☐ The MED is exchanged between autonomous systems.
- ☐ The MED is local to an AS.

1. 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: "Addressing BGP Communities")

- ☐ The MED attribute is designed to influence outbound path selection only.
- ☐ The AS path attribute would be used for path selection regardless of any configured MED value.
- ☐ The weight attribute will always be used, given that it is first in the BGP route selection process.
- ☐ The MED cannot be propagated across several autonomous systems.

1. Does the community attribute have any influence on BGP path selection? (Source: "Addressing BGP Communities")

☐ No, communities are simply tags that are applied to BGP routes.

☐ No, communities are nontransitive attributes.

☐ Yes, BGP paths are selected based on the value in the community tag.

☐ Yes, the community attribute is part of the BGP route selection process.

1. Match the steps with the actions describing how BGP communities can facilitate proper return path selection. (Source: "Addressing BGP Communities")

|                                                                                                                                                                                    |        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Define administrative policy goals that you need to implement.                                                                                                                     | Step 3 |
| Enable community distribution throughout your AS to allow community propagation.                                                                                                   | Step 5 |
| Define the filters and route selection policy that will achieve the required goals.                                                                                                | Step 2 |
| Apply communities on incoming updates from neighboring autonomous systems or tell the neighbors to set the communities themselves.                                                 | Step 1 |
| 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. | Step 4 |
| Assign a community value to each goal.                                                                                                                                             | Step 6 |

1. How many community tags can be attached to a single BGP route?  
(Source: "Addressing BGP Communities")

1

32

256

It depends on the number that is configured with the **ip bgp community** command.

1. Which three of the following are activities that are required to successfully deploy BGP communities in a BGP-based network? (Choose three.) (Source: "Addressing BGP Communities")

- ☒ Setting communities, which requires a route-map.
- ☒ Creating community-lists to be used within route-maps to match on community values.
- ☒ Enabling community propagation per neighbor for designated internal neighbors.
- ☐ Creating route-maps where community-lists are used to match on community values.



1. Which two of the following statements about the Cisco IOS commands that are required to configure route tagging with BGP communities are accurate? (Choose two.) (Source: "Addressing BGP Communities")

☐ You can attach up to 35 communities to a single route with one route-map set statement.

☐ Omitting the **additive** keyword from the **set community** command results in overwriting any original community attributes.

☐ The global command **ip bgp-community new-format** is recommended on all routers whenever communities contain the AS number.

☐ You cannot use a route-map with redistribution from another routing protocol.

1. Which of the following statements about the Cisco IOS command that is required to enable BGP community propagation to BGP neighbors is accurate? (Source: "Addressing BGP Communities")

☐ Community propagation to BGP neighbors is automatically configured.

☐ It is necessary to manually strip communities in outgoing BGP updates.

☐ The **neighbor ip-address send-community** command cannot be applied to a peer group.

☒ The **neighbor ip-address send-community** command is needed to propagate community attributes to BGP neighbors.

1. Which match criteria are specified in a standard BGP community-list? (Source: "Addressing BGP Communities")

☐ destination IP addresses

☐ regular expressions

☐ community attribute values

☐ AS numbers

1. What is the result of tagging a route with the no-export community?  
(Source: "Addressing BGP Communities")

- The route will not be advertised within the local AS.
- The upstream AS will not be allowed to export the route.
- The route cannot be exported to another routing protocol.
- The router will not propagate the route to any external neighbors except to intra-confederation external neighbors.

1. Match the functions to the Cisco IOS commands that monitor BGP communities. (Source: "Addressing BGP Communities")

Displays all routes in a BGP table that have all the specified communities attached.

**show ip bgp community**

Displays all routes in BGP table that have exactly the specified communities attached.

**show ip bgp community-list *clist***

Displays all routes in a BGP table that have at least one community attached.

**show ip bgp community as:nn [as:nn ...]**

Displays all routes in BGP table that match community-list *clist*.

**show ip bgp community as:nn [as:nn ...]  
exact**

1. Which two of the following statements about the function of the BGP Link Bandwidth feature are accurate? (Choose two.) (Source: "Addressing BGP Communities")

- ☐ The BGP Link Bandwidth feature is used to enable multipath load balancing for external links with unequal bandwidth capacity.
- ☐ When the BGP Link Bandwidth feature is enabled, routes learned from directly connected external neighbor are propagated through the IBGP network with the bandwidth of the source external link.
- ☐ The configuration of the BGP Link Bandwidth feature can be used only with IBGP multipath features to enable unequal-cost load balancing over multiple links.
- ☐ To configure BGP to advertise the bandwidth of links that are used to exit an AS use the **bgp dmzlink-bw** command.

1. Which three of the following functions of BGP named community-lists are accurate? (Choose three.) (Source: "Addressing BGP Communities")

- ☐ Allows the network operator to assign meaningful names to community-lists
- ☐ Sets limits on the number of named community-list that can be configured
- ☐ Cannot be configured with regular expressions and with numbered community- lists.
- ☐ Increases the number of community-lists that can be configured by a network operator.
- ☐ Applies policies to large numbers of routes by using match and set clauses in the configuration of route maps.
- ☐ Sets limit of 200 community groups that can be configured within each type of list.

1. Which two of the following statements about the BGP Cost Community feature are accurate? (Choose two.) (Source: "Addressing BGP Communities")

- ☐ The path with the highest cost community number is preferred.
- ☐ The cost community attribute influences the BGP best path selection process at the POI.
- ☐ The cost community can be used as a "tie breaker" during the best-path selection process.
- ☐ If the cost community values are equal, then cost community comparison proceeds to the next highest community ID for this POI.



1. Which three of the following are functions of the BGP Support for Sequenced Entries in Extended Community Lists? (Choose three.)  
(Source: "Addressing BGP Communities")

- ☐ 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 an extended community-list to use custom values.
- ☐ Removes or resequences extended community-list entries while deleting the entire existing extended community-list.
- ☐ Is activated by the **ip extcommunity-list** command in global configuration mode.
- ☐ Configures sequence numbers for standard community-list entries.

## Answer Key

1. Number the 10 BGP selection criteria in the order in which they are used, from the first to the last, when selecting the BGP path that is submitted to the IP routing table. (Source: "Influencing BGP Route Selection with Weights")

|                                                           |    |
|-----------------------------------------------------------|----|
| Prefer the highest weight.                                | 1  |
| Prefer the shortest AS path.                              | 4  |
| Prefer an EBGp path over an IBGP path.                    | 7  |
| Prefer the highest local preference.                      | 2  |
| Prefer the lowest origin code (IGP < EGP < incomplete).   | 5  |
| Prefer the path through the closest IGP neighbor.         | 8  |
| Prefer the route that was originated by the local router. | 3  |
| Prefer the path with the lowest neighbor BGP router ID.   | 10 |
| Prefer the oldest route for EBGp paths.                   | 9  |
| Prefer the lowest MED.                                    | 6  |

1. What is the default weight for routes that are received from a BGP neighbor? (Source: "Influencing BGP Route Selection with Weights")

0

100

32768

It depends on the Cisco IOS release.

1. When are the weights that are configured on a neighbor enforced?  
(Source: "Influencing BGP Route Selection with Weights")

- Before the new weights can take effect, the BGP process on the router must be removed and reconfigured.

- The router must first be rebooted for the new weights to take effect.

- The new weights will be applied after the BGP update interval of 30 minutes expires.

- The new weight configuration is applied to all routes that are received following the configuration change.

1. If you want all routes received from the neighbor to have the new weight value applied, you have to reestablish BGP session. True or false? (Source: "Influencing BGP Route Selection with Weights")

☐ true

☐ false

1. How could you implement a primary/backup ISP routing policy by using weights? (Source: "Influencing BGP Route Selection with Weights")

- Assign higher weights to all routes that are received from the backup ISP.

- Assign lower weights to all routes that are received from the backup ISP.

- Assign higher weights to all routes that are received from the primary ISP.

- Assign lower weights to all routes that are received from the primary ISP.

1. When you are using route-maps to modify weights, what happens by default to a route that does not match any of the route-map statements? (Source: "Influencing BGP Route Selection with Weights")

☐ The route is accepted with the weight attribute unmodified.

☐ The route is discarded.

☐ The route is inserted into the BGP table but not into the IP routing table.

☐ An error is displayed on the router console and in router debugs.

1. Which method of influencing route selection with weights is the last to be applied on an incoming interface? (Source: "Influencing BGP Route Selection with Weights")

☐ prefix-list

☐ route-map

☐ filter-list weight

☐ default weight



1. You can set the weight using a route map applied to a neighbor in the outgoing direction. True or false? (Source: "Influencing BGP Route Selection with Weights")

☒ true

☐ false

1. Which two of the following statements are correct regarding BGP route selection? (Choose two.) (Source: "Influencing BGP Route Selection with Weights")

- ☐ If two routes have the same weight attribute, the route with the lowest local preference is chosen.
- ☒ The route with the highest weight is always chosen first.
- ☐ The weight attribute is global within an AS.
- ☒ The weight attribute is only local to the local router.
- ☐ The weight value is propagated by all routers.

1. What is a key difference between the local preference and weight attributes? (Source: "Setting BGP Local Preference")

- Local preference is local to the route on which it is configured.
- Local preference is local to the AS within which it has been configured.
- Local preference is local to the BGP administrative domain.
- Local preference is global to a BGP domain.

1. What is the default local preference for routes that are received from a BGP neighbor? (Source: "Setting BGP Local Preference")

0

100

There is no default local preference value.

It depends on the Cisco IOS release.

1. Which two statements regarding local preference are true? (Choose two.) (Source: "Setting BGP Local Preference")



The higher value for local preference is preferred.



Local preference is used only between EBGp neighbors



The lower value for local preference is preferred.



Local preference is used only between IBGP neighbors

1. Which two of the following statements about the influence of local preference on BGP route selection is accurate? (Choose two.)  
(Source: "Setting BGP Local Preference")

☐ When you set local preference, you can view it on neighboring routers, but you must reset it.

☒ You can use local preference to ensure AS-wide route selection policy.

☐ Local preference is used to select routes with unequal weights.

☒ Local preference is the second-strongest criterion in the route selection process.

1. Let say we have two routes to the same destination, fist one has a weight of 50 and a local preference of 150, while the other one has a weight of 150 and a local preference of 50. The fist route will be selected as the best one, since it has higher local preference value. True or false? (Source: "Setting BGP Local Preference")

☒ true

☐ false

1. Which Cisco IOS command is used to change the default value of local preference? (Source: "Setting BGP Local Preference")

☐ **set local-preference**

☐ **bgp default local-preference**

☐ **show ip bgp**

☐ **show ip bgp *prefix***



1. Which Cisco IOS command is used to configure BGP local preference with route-map statements? (Source: "Setting BGP Local Preference")

**set local-preference**

**bgp default local-preference**

**show ip bgp**

**show ip bgp *prefix***

1. Locally applied default value can be displayed with **show ip bgp** command. True or false? (Source: "Setting BGP Local Preference")

true

false

1. Which Cisco IOS command is necessary to display the locally applied BGP value? (Source: "Setting BGP Local Preference")

☐ **show bgp preference detail**

☐ **show ip bgp**

☐ **show ip bgp detail**

☐ **show ip bgp *prefix***

1. Which two statements regarding the AS path are true? (Choose two.) (Source: "Using AS Path Prepending")

- ☒ The shorter AS path is preferred.
- ☐ The longer AS path is preferred.
- ☒ The AS path is prepended and exchanged between autonomous systems.
- ☐ The AS path is local to an AS.

1. What is AS path prepending? (Source: "Using AS Path Prepending")

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.

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.

When a router, sending a BGP update, adds its AS number to the AS path attribute multiple times.

When a router uses the AS path attribute in route selection.

1. The AS path will be the route selection criterion that is used when which of the following is true? (Source: "Using AS Path Prepending")

- ☐ It is the first criterion that is used in BGP route selection.
- ☐ It is used when there is no difference in weight, local preference, or route origination.
- ☐ It is used when the MED is identical on the candidate routes.
- ☐ The weight, local preference, MED, and origin attributes must be identical before the AS path attribute is used for route selection.

1. Which command do you use to manipulate the AS path attribute?  
(Source: "Using AS Path Prepending")

☐ The global configuration command **set as-path prepend as-number**.

☐ The router configuration command **set as-path prepend as-number**.

☐ The **set as-path prepend as-number** command in a route-map.

☐ The interface global command **set as-path prepend as-number**.

1. The configuration below is from a router in AS 347, which is advertising network 11.0.0.0/8 to an EBGP neighbor 2.0.0.2 in AS 529. What are the contents of the AS path attribute for route 11.0.0.0/8 on a router that is residing in AS 529? (Source: "Using AS Path Prepending")
- ```
route-map addAS permit 10
set as-path prepend 347 347 347
router bgp 347
neighbor 2.0.0.2 remote-as 529
neighbor 2.0.0.2 route-map addAS out
```

347 347 347

347 347 347 347

529 347 347 347

529 347 347 347 347

1. Why do you need to use AS path prepending? (Source: "Using AS Path Prepending")

AS path prepending allows a customer to potentially influence return path route selection.

AS path prepending is used on a customer router to control outgoing route updates.

Service providers use AS path prepending to control incoming updates from a customer AS.

AS path prepending is used between service providers that are connected to the same customer AS to determine which will be the primary link to the customer.

1. Which regular expression will allow any AS path number, but with a length of one? (Source: "Using AS Path Prepending")

☐ `^[0-9]+$`

☐ `^[0-9](_[0-9])*$`

☐ `^_[0-9]$`

☐ `^[1-8]+$`

1. How does AS path prepending affect a router? (Source: "Using AS Path Prepending")

AS path prepending is simply a term that is used to describe when a router uses the AS path attribute in route selection and hence does not affect router resources.

The longer the AS path attribute attached to BGP updates, the more router memory requirements increase.

AS path prepending does not impact the router because Cisco IOS software recognizes that AS path prepending is in use and stores a single AS number with a pointer to the number of AS path prepends.

AS path prepending causes the router to operate in process-switching mode because the BGP update must be stored, manipulated, and then rewritten to accommodate the new AS path attribute.

1. Which two of the following are characteristics of the function of the BGP Hide Local-Autonomous System feature? (Choose two.)
(Source: "Using AS Path Prepending")

- ☒ Allows you to transparently change the AS number for the entire BGP network.
- ☒ Ensures that routes can be propagated throughout the AS.
- ☐ Allows customization of the AS number for EBGp peer groupings through the **set as-path** command.
- ☐ Changes the AS number for all IBGP peers at the same time.

1. What is the typical application of the MED attribute? (Source: "Understanding BGP Multi-Exit Discriminators")

- To influence path selection out of an originating AS.

- To provide a strong metric to select the best path when multiple routes exist.

- To have a BGP attribute traversing many autonomous systems while influencing path selection.

- To influence the return path of traffic back into an AS.

1. What is the default MED value? (Source: "Understanding BGP Multi-Exit Discriminators")

0

100

There is no default MED value.

It depends on the Cisco IOS release.

1. What are three BGP attributes that are compared before the MED? (Choose three.) (Source: "Understanding BGP Multi-Exit Discriminators")

☒ largest weight

☒ originated routes

☒ AS path length

☐ lowest IP address

1. Which two statements about the Cisco IOS command that is required to configure changes to the default BGP MED on a Cisco IOS router are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")

☐ The MED is a mandatory attribute.

☐ Using the **default-metric** command in BGP configuration mode will cause one redistributed network to have the specified MED value.

☒ There is no MED attribute that is attached to a route by default.

☒ To set the default metric value (MED) for BGP routes, use the default-metric command.

1. Which two statements about the Cisco IOS commands that are required to configure changes to the BGP MED attribute with route-map statements are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")

☒ The **set metric** command is used within route-map configuration mode to set the MED attribute.

☐ The **neighbor address route-map name in | out** command applies a route-map to incoming updates from all neighbors.

☐ Per-neighbor MED is configured by using a route-map with a match condition.

☒ You can use a route-map to set the MED on incoming or outgoing updates.

1. Which two of the following statements about the Cisco IOS commands that are required to configure advanced MED features on Cisco routers are accurate? (Choose two.) (Source: "Understanding BGP Multi-Exit Discriminators")



The **bgp bestpath med confed** command allows routers to compare paths learned from confederation peers.



When you enable a deterministic MED comparison, you allow a router to compare MED values after it considers BGP route type (external or internal) and IGP metric to the next-hop address.



Use the **bgp always-compare-med** command to force the router to compare the MED even if the paths come from different autonomous systems.



Cisco IOS software, on the other hand, regards a missing MED attribute as having a value of 1.

1. If you configure inbound soft reconfiguration with a route-map and issue the **show ip bgp prefix** command, which value of the MED attribute is displayed? (Source: "Understanding BGP Multi-Exit Discriminators")

☐ Only the original route (no MED) is displayed.

☐ Both the original route and the modified route are displayed.

☐ Only the modified route is displayed.

☐ The MED attribute is not displayed with the **show ip bgp prefix** command.

1. Which of the following statements about the Cisco IOS commands that are required to troubleshoot BGP MED configurations on a Cisco router is accurate? (Source: "Understanding BGP Multi-Exit Discriminators")

To see the original MED, you need to enable hard reconfiguration on the router.

The command **show ip bgp neighbor *address* received-routes** displays the original updates before any filters or route-maps have filtered or changed them.

If hard reconfiguration is enabled, the original updates to the MED attribute are available by using the **show ip bgp *prefix*** command.

Issuing the **show ip route** command will display the MED value.

1. Which two statements regarding the MED are true? (Choose two.)
(Source: "Understanding BGP Multi-Exit Discriminators")

☐	The higher value for the MED is preferred.
☒	The lower value for the MED is preferred.
☐	The MED is exchanged between autonomous systems.
☒	The MED is local to an AS.

1. 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: "Addressing BGP Communities")

- ☐ The MED attribute is designed to influence outbound path selection only.
- ☒ The AS path attribute would be used for path selection regardless of any configured MED value.
- ☐ The weight attribute will always be used, given that it is first in the BGP route selection process.
- ☒ The MED cannot be propagated across several autonomous systems.

1. Does the community attribute have any influence on BGP path selection? (Source: "Addressing BGP Communities")

- ☐ No, communities are simply tags that are applied to BGP routes.
- ☐ No, communities are nontransitive attributes.
- ☐ Yes, BGP paths are selected based on the value in the community tag.
- ☐ Yes, the community attribute is part of the BGP route selection process.

1. Match the steps with the actions describing how BGP communities can facilitate proper return path selection. (Source: "Addressing BGP Communities")

Define administrative policy goals that you need to implement.	Step 1
Define the filters and route selection policy that will achieve the required goals.	Step 2
Assign a community value to each goal.	Step 3
Apply communities on incoming updates from neighboring autonomous systems or tell the neighbors to set the communities themselves.	Step 4
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.	Step 5
Enable community distribution throughout your AS to allow community propagation.	Step 6

1. How many community tags can be attached to a single BGP route?
(Source: "Addressing BGP Communities")

1

32

256

It depends on the number that is configured with the **ip bgp community** command.

1. Which three of the following are activities that are required to successfully deploy BGP communities in a BGP-based network? (Choose three.) (Source: "Addressing BGP Communities")

- ☒ Setting communities, which requires a route-map.
- ☒ Creating community-lists to be used within route-maps to match on community values.
- ☐ Enabling community propagation per neighbor for designated internal neighbors.
- ☒ Creating route-maps where community-lists are used to match on community values.

1. Which two of the following statements about the Cisco IOS commands that are required to configure route tagging with BGP communities are accurate? (Choose two.) (Source: "Addressing BGP Communities")

☐ You can attach up to 35 communities to a single route with one route-map set statement.

☒ Omitting the **additive** keyword from the **set community** command results in overwriting any original community attributes.

☒ The global command **ip bgp-community new-format** is recommended on all routers whenever communities contain the AS number.

☐ You cannot use a route-map with redistribution from another routing protocol.

1. Which of the following statements about the Cisco IOS command that is required to enable BGP community propagation to BGP neighbors is accurate? (Source: "Addressing BGP Communities")

☐ Community propagation to BGP neighbors is automatically configured.

☐ It is necessary to manually strip communities in outgoing BGP updates.

☐ The **neighbor ip-address send-community** command cannot be applied to a peer group.

☒ The **neighbor ip-address send-community** command is needed to propagate community attributes to BGP neighbors.

1. Which match criteria are specified in a standard BGP community-list? (Source: "Addressing BGP Communities")

☐ destination IP addresses

☐ regular expressions

☐ community attribute values

☐ AS numbers

1. What is the result of tagging a route with the no-export community?
(Source: "Addressing BGP Communities")

The route will not be advertised within the local AS.

The upstream AS will not be allowed to export the route.

The route cannot be exported to another routing protocol.

The router will not propagate the route to any external neighbors except to intra-confederation external neighbors.

1. Match the functions to the Cisco IOS commands that monitor BGP communities. (Source: "Addressing BGP Communities")

Displays all routes in a BGP table that have at least one community attached.

show ip bgp community

Displays all routes in a BGP table that have all the specified communities attached.

show ip bgp community as:nn [as:nn ...]

Displays all routes in BGP table that have exactly the specified communities attached.

**show ip bgp community as:nn [as:nn ...]
exact**

Displays all routes in BGP table that match community-list *clist*.

show ip bgp community-list *clist*

1. Which two of the following statements about the function of the BGP Link Bandwidth feature are accurate? (Choose two.) (Source: "Addressing BGP Communities")

☒ The BGP Link Bandwidth feature is used to enable multipath load balancing for external links with unequal bandwidth capacity.

☒ When the BGP Link Bandwidth feature is enabled, routes learned from directly connected external neighbor are propagated through the IBGP network with the bandwidth of the source external link.

☐ The configuration of the BGP Link Bandwidth feature can be used only with IBGP multipath features to enable unequal-cost load balancing over multiple links.

☐ To configure BGP to advertise the bandwidth of links that are used to exit an AS use the **bgp dmzlink-bw** command.

1. Which three of the following functions of BGP named community-lists are accurate? (Choose three.) (Source: "Addressing BGP Communities")

- ☒ Allows the network operator to assign meaningful names to community-lists
- ☒ Sets limits on the number of named community-list that can be configured
- ☐ Cannot be configured with regular expressions and with numbered community- lists.
- ☒ Increases the number of community-lists that can be configured by a network operator.
- ☒ Applies policies to large numbers of routes by using match and set clauses in the configuration of route maps.
- ☐ Sets limit of 200 community groups that can be configured within each type of list.

1. Which two of the following statements about the BGP Cost Community feature are accurate? (Choose two.) (Source: "Addressing BGP Communities")

- ☐ The path with the highest cost community number is preferred.
- ☒ The cost community attribute influences the BGP best path selection process at the POI.
- ☒ The cost community can be used as a "tie breaker" during the best-path selection process.
- ☐ If the cost community values are equal, then cost community comparison proceeds to the next highest community ID for this POI.

1. Which three of the following are functions of the BGP Support for Sequenced Entries in Extended Community Lists? (Choose three.)
(Source: "Addressing BGP Communities")

- ☒ 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 an extended community-list to use custom values.
- ☐ Removes or resequences extended community-list entries while deleting the entire existing extended community-list.
- ☒ Is activated by the **ip extcommunity-list** command in global configuration mode.
- ☐ Configures sequence numbers for standard community-list entries.

Customer-to-Provider Connectivity with BGP

Introduction

Customers who are connecting to a service provider must consider various requirements and choose the proper connectivity type. In general, three types of connectivity are available: connecting to a single service provider, connecting to a single service provider using two links, and connecting to two independent service providers. Based on the connectivity type the proper routing methods, [IP addressing](#), and [AS](#) number have to be chosen. After all of these input parameters are known the customer and service provider can implement the desired connectivity type.

Upon completing this module, you will be able to:

- Describe the requirements to connect customer networks to the Internet in a service provider environment
- To implement customer connectivity by using static routing
- Defines how to implement Multihomed customer connectivity using BGP to a single ISP
- Define how to implement Multihomed customer connectivity using BGP to multiple ISPs

Understanding Customer-to-Provider Connectivity Requirements

Overview

Customers connect to the Internet by using service providers to enable applications such as intranet connectivity with [VPNs](#), extranet connectivity with suppliers, and other Internet applications. When planning network connectivity to an [ISP](#), you as a network designer must give careful consideration to the various aspects of the connectivity. These aspects include physical connection types, the redundancy that the chosen connection method provides, [IP addressing](#) requirements, and [AS](#) numbering considerations, if the network design is going to meet both business and technical requirements of the applications that are planned for the network.

In this lesson, you will discuss solutions for connecting customer networks to service providers. You will also discuss customer network redundancy requirements, routing requirements, IP addressing requirements, and AS numbering requirements.

Upon completing this lesson, you will be able to:

- Identify various physical connections that the customers use to connect to a service provider
- Describe the levels of redundancy that is provided by each physical connection type that the customers use to connect to a service provider
- Identify various routing schemes that the customers use to connect to a service provider
- Describe routing schemes that are appropriate for each physical connection type that the customers use to connect to a service provider
- Describe the addressing schemes that the customers use to connect to a service provider
- Describe the AS numbering schemes that the customers use to connect to a service provider

Customer-to-Provider Connectivity Types

Internet customers have a wide range of connectivity and redundancy requirements:

- **Single-homed customers:**
 - Single permanent connection to a single service provider.
 - Multiple permanent connections to a single provider in primary and backup configuration.
 - Multiple permanent connections to a single provider used for load sharing of traffic.
- **Multihomed customers:**
 - Multiple permanent connections to multiple service providers for maximum redundancy.

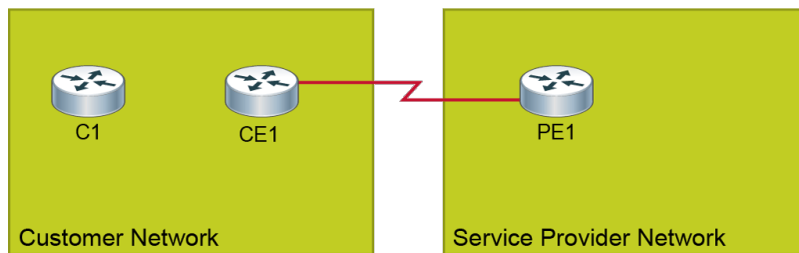
Service provider customers have different requirements, such as redundancy, stability, security, flexibility, for their Internet connectivity. These different requirements result in different solutions:

- A single permanent connection to one [ISP](#). This solution meets the requirements for most of the customers.
- Multiple permanent connections to one ISP, in which one of the lines is primary and the other line is used for backup only. This setup also provides redundancy on the links. Compared to a dial-up backup, a permanent backup link is preferred for various reasons. For example, the severe bandwidth limitations on dial-up lines and the time that is required to establish a dial-up connection.
- Multiple permanent connections to one ISP, which is used for load sharing of traffic. This solution gives redundancy on the links but also provides extra bandwidth.
- Multiple permanent connections to more than one ISP. This solution provides the highest level of redundancy, because not only does it cope with link-level failures but also with failures within the network of a service provider.

Customer Redundant Connectivity

The level of redundancy depends on a physical connection type that the customers use to connect to a service provider.

Single Permanent Connection



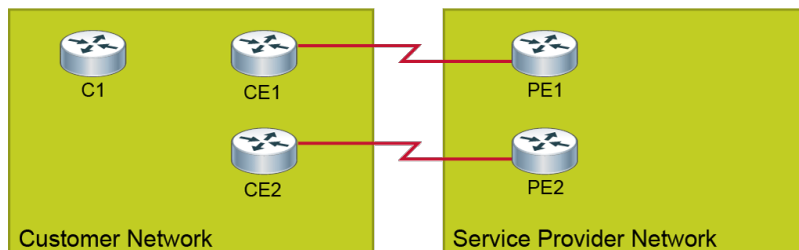
- The simplest setup is a single link between the customer network and the service provider.
- There is no redundancy for link, equipment, or service provider failure.

A single permanent connection to one [ISP](#) is the most common setup. This setup is also the simplest to implement.

The customer connects to a single service provider using a single link. The customer has an edge router that is connected to the edge routers of the ISP. This connection is permanent and could be a leased line, a [Frame Relay](#), or [ATM PVC](#), [DSL](#), [FETx](#), cable, or something equivalent.

You do not have redundancy in this solution. Any failure on the permanent link or either of the two edge routers causes a complete outage of the service. Serious failures within the ISP network that affect all customers of this ISP also affect the customer in this example.

Multiple Permanent Connections Providing Redundancy



To increase redundancy, customers install several physical links to the service provider.

- Redundant links are used as follows:
 - Primary and backup.
 - For load sharing.
- Redundancy is for link or equipment failure.
- There is no redundancy for service provider failure.

In this setup, you have one CE router that is connected to one PE router, and another CE that is connected to another PE router. If one of these routers fails, only one of the connections breaks down; the other connection is still available.

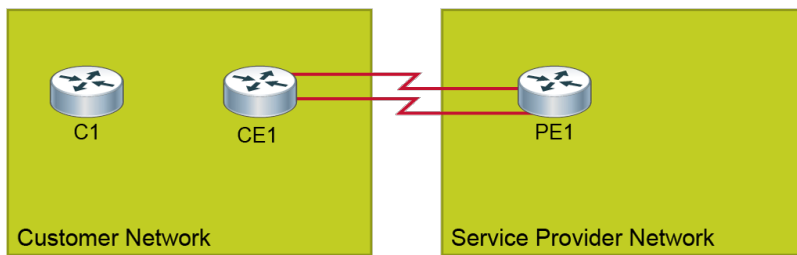
These two links may be implemented in two scenarios:

- **Primary and backup:** One link is always active, while another one is the backup and is used only when the primary link fails. Backup PVCs in Frame Relay or ATM networks can be very cost-efficient in case that these PVCs carry only a very small volume of traffic and that the primary path is available.
- **Load sharing:** Both links are used at the same time to provide load balancing. The distribution of the load over the links is more complicated than when both links terminate in the same router.

Again, because the customer is connected to a single ISP, redundancy is provided for link, and customer, or ISP device failure. The failures in the ISP network will still

affect the customer. This solution can mitigate router and link failures, but it cannot mitigate ISP failures.

Multiple Permanent Connections Providing Load Sharing



Customers wanting to increase their access speed can install several physical links between a pair of routers.

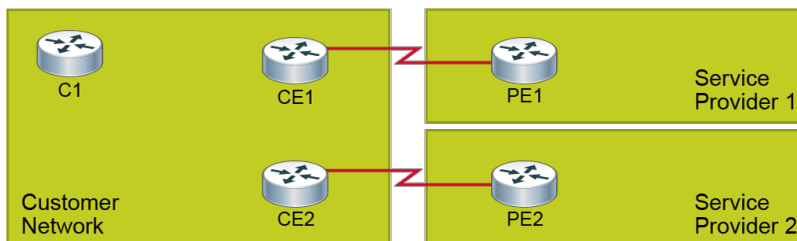
- There is redundancy for link failure.
- There is no redundancy for equipment failure.
- Load sharing in this setup is optimal.

In this setup, you have a single router in the customer network that is connected to a single router in the ISP network. The redundancy is limited to the link level because router failures are not covered. Using two parallel links between two routers allows for an optimal distribution of load over the links.

Depending on the switching path that is used in the customer and the ISP routers, load sharing can be performed based on various options. On the destination address only (fast switching), on source-destination address pairs (default behavior for CEF), or on a packet-by-packet basis (process switching or CEF).

As in the previous examples, because the customer is connected to a single ISP, serious ISP network failure will affect the customer regardless of the link backup.

Connections to Multiple Service Providers



Customers with maximum redundancy requirements install physical links to multiple service providers.

- Redundant links are used as follows:
 - Primary and backup.
 - Primary and backup with direct traffic.
 - For load sharing.
- There is redundancy for link, equipment, or service provider failure.

In the example, two **CE** routers have one permanent connection each to a different ISP. Link and device failures are covered in the same way as in the previous examples. However, because the two connections in this example go to two different ISPs, the redundancy also covers problems within one ISP network. If one service provider fails, traffic will still flow through another service provider, and the highest level of redundancy is achieved.

In this setup, three scenarios are possible:

- **Primary and backup:** One link is always active, while another one is the backup and is used only when the primary link fails.
- **Primary and backup with direct traffic to the backup service provider:** One link is active, and another one is the backup. Except for the traffic that should go directly to service provider 2.
- **Load sharing:** Both links are used at the same time to provide load balancing.

In all scenarios, the customer has to cooperate with a service provider to achieve the desired setup. Load sharing between the links can never be optimal. Equal distribution of the return traffic load from the Internet over the two separate links cannot be achieved. Distribution of the load of outgoing traffic is done based on destination addresses. Slowly adjusting the router configuration parameters and observing the link traffic load changes can help you achieve an acceptable distribution of traffic between two links.

Avoiding any load on the backup link may require assistance from the ISP to which the backup link is connected.

Customer-to-Provider Routing Schemes

Different solutions for connecting a customer network to the network of an [ISP](#) require different methods of routing information exchange.

- Static or dynamic routing can be used between an Internet customer and a service provider.
- BGP is the only acceptable dynamic routing protocol.
- Because of its lower complexity, static routing is preferred where possible.

Routing options:

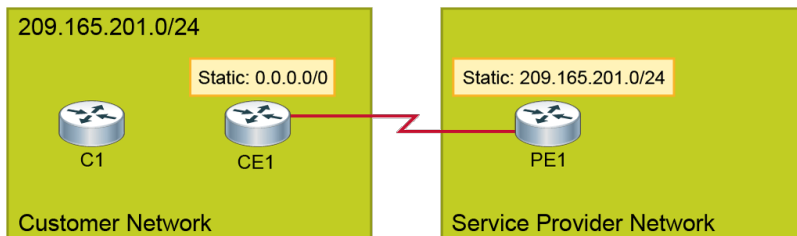
- **Static routing:** Static routing is preferred because of its lower complexity. In a normal case, the customer network must have a default route to the ISP network and the ISP network must have a route to the IP prefixes that the customer has in its network. As always, static routing provides very low redundancy, if any.
- **Dynamic routing:** Dynamic routing provides redundancy. The customer and the ISP networks must be configured to exchange a common routing protocol. [BGP](#) is the only choice because of the large volumes of routing information, the inherent security mechanisms of BGP, and the ability of BGP to handle routing policies.

Customer Routing Schemes

Different routing schemes are appropriate for each physical connection type that the customers use to connect to a service provider.

Routing Scheme: Customer with a Single Connection to a Service Provider

When the customer has a single connection to the service provider, static routing is usually adequate. The physical topology does not provide any redundancy, and it is therefore unnecessary to add the complexity of dynamic routing.



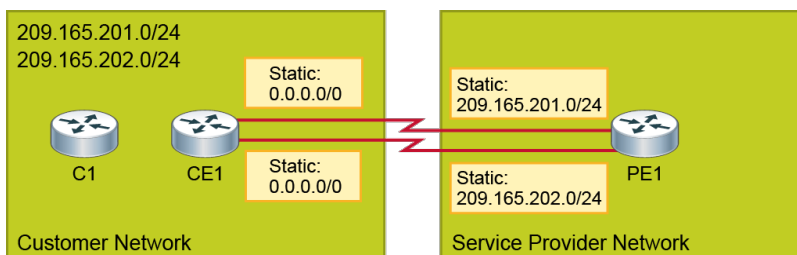
Customers with a single connection to a service provider typically do not require BGP:

- Static routing is usually adequate.
 - The static route for the provider-assigned address space of the customer is on the PE router.
 - The static default route is on the CE router.
- BGP should not be used in this setup.

In this case, keep the network simple by avoiding the use of [BGP](#). If there is static routing, a default route is needed on the [CE](#) router that points to the [PE](#) router. A static route for the customer network is needed on the PE router, which points to the CE router.

If you use BGP, the service provider sends a default route to the customer, and the customer advertises its own address space to the service provider.

Routing Scheme: Customer with Multiple Connections to a Service Provider

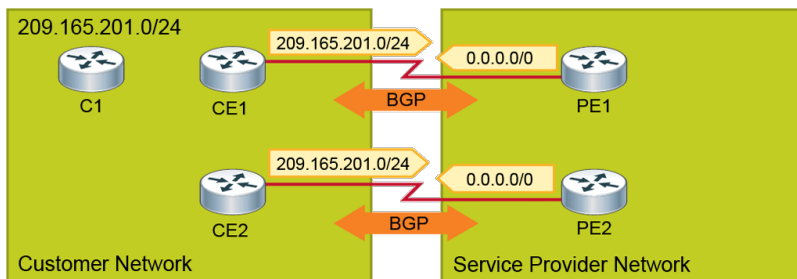


- Static routing is preferred if physical link failure can be detected.
- Traffic will enter a "black hole" if the physical link failure is not detected.

Multiple permanent connections between a single router on the customer network and a single router on the service provider network should be configured with static routing. This configuration enables the link-level procedures to detect a link failure.

With this type of connection, two default routes are configured on the CE router that point to the PE router. Two static routes are configured on PE router, pointing to both links between the customer and the [ISP](#). If either of the links fails, the link-level procedures should detect this failure and place the interface in a down state. In this case, the static route is invalid and is not used for forwarding packets. The router will then forward all packets over the remaining link.

If the link-level procedures cannot detect a link failure, the static route pointing out over the failed link is still valid. The router continues to use this static route to send some of the traffic out on the failed interface. This situation effectively creates a "black hole" for some of the traffic.

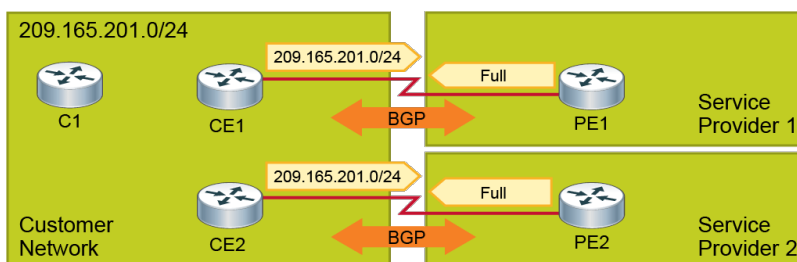


- You can still use static routing if link and service provider equipment failure can be detected.
- BGP between the customer and the service provider is usually used in this setup.
 - The service provider originates the default route (in the primary or secondary scenario) or specific routes (in the load-balancing scenario).
 - The customer originates its address space.

You can also use static routing for multiple permanent connections between two different routers on the customer network to two different routers on the service provider network. However, you can use static routing only if the link-level procedures can detect link failures. When one of the connections is lost, the link-level procedure detects this loss and places the interface in a down state. Because the interface is in the down state, the static route that points out of the down interface becomes invalid. As a result, the customer's router stops the redistribution of the default route into the customer [IGP](#). The provider's router stops the redistribution of the static route, that points the customer network, into the service provider BGP.

However, customers that require the use of multiple connections and multiple routers very often do not rely on the link-level procedures. A better option to detect link failures is using BGP between CE and PE routers. Because BGP uses handshaking and reliable transfer, it always detects a failed link or failed remote router. BGP can also be used to achieve a primary or secondary scenario or load balancing of traffic over both links.

Routing Scheme: Customer with Connections to Multiple Service Providers



- Static routing is not possible.
- BGP must be used:
 - The service provider originates the default route (in the primary or secondary scenario), the default route, and service provider-owned routes (in the primary or secondary scenario with different traffic scenario), or all routes (in the load-balancing scenario).
 - The customer originates its address space.

Multiple permanent connections to more than one ISP always require the use of dynamic routing with BGP. The customers that require this type of connection do not want only to protect the network connectivity from link failures or remote router failures. They also want to protect their network connectivity from serious problems in the ISP network.

Monitoring the link status cannot detect a problem inside one of the ISP networks. If the link is still up and the ISP edge router is still up, the link-level procedures do not indicate any problems. However, the ISP network may suffer from severe problems. An ISP network can be partitioned or disconnected from the rest of the Internet without having any problems with the edge router and the access line to the

customer network.

The only way to detect this situation is to use BGP with both ISPs and receive full Internet routing from both of them. When one of the ISPs has problems, the edge router, being the BGP neighbor of the customer, withdraws the routes that it can no longer reach. This action means that the customer routers know which Internet routes each ISP can reach at the moment.

CE routers advertise the customer network to the service provider. PE routers advertise only the default route or the default route plus the service provider-owned networks or all routes to the CE routers. The level of routes that the service provider advertises is determined based on the required customer scenario.

Customer Addressing Requirements

Customers who are connected to a single service provider usually have their address space assigned by the service provider, which is known as [PA addresses](#). Customers that are connected to more than one [ISP](#) should, if possible, assign their own address space and not have addresses that are delegated from any of their ISPs. Such assigned addresses are called [PI addresses](#).

Single-homed customers:

- Customer gets the address space from the provider—PA address space.
- Single (link) IP address or multiple (subnet) IP address.
- If service provider changes, readdressing is needed.
- Inside the customer network, private addresses are used, with a combination of NAT.

Multihomed customers:

- Customer assigns its own address space—PI address space
- Customer has to advertise address space to all service providers.
- If service provider changes, readdressing is not needed.

Single-Homed Customers: Customers that are connected to a single ISP usually get their address space assigned by the ISP. Customers can either request a single [IP address](#), which is used on the access link as well, or they can request an entire IP subnet. A service provider is usually assigned a large address space to delegate to its customers. All customers of one ISP get their addresses from one address space or a few address spaces. It is very likely that the ISP is able to aggregate the customer addresses before sending the routes to the rest of the Internet.

If the customer should decide to change its service provider, the customer must return its PA addresses to the old ISP. The customer then receives a new assignment of PA addresses from the new ISP. Otherwise, the ISPs are no longer able to perform efficient address aggregation. The consequence for the customer is that the customer has to renumber its network when it changes its service provider.

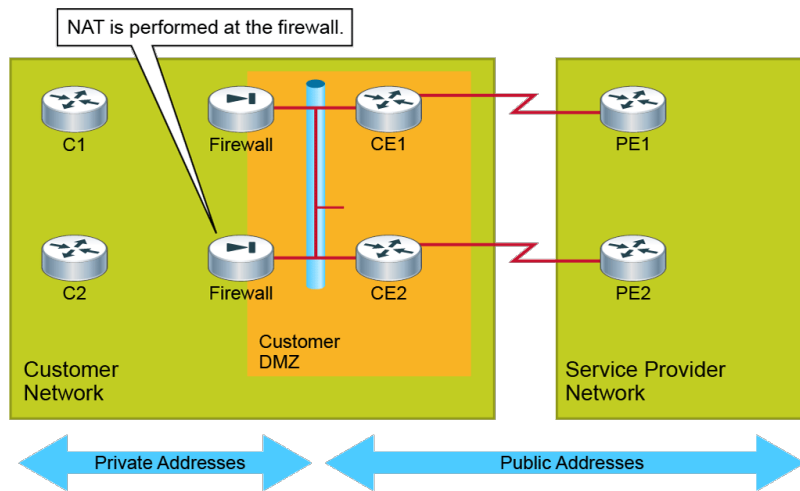
Multihomed Customers: Customers that are connected to multiple ISPs usually assign their own address space. None of the service providers advertises the address, and the customer has to advertise it to both service providers. This arrangement means that no ISP can aggregate the customer routes before sending them to the rest of the Internet. The routes propagate through the Internet with the prefix lengths given.

Some large ISPs filter out routes with long prefixes. ISPs do not want to populate their routing tables with many explicit routes that should have been aggregated into a route summary before they were sent to them. As a result, the customer announcing small blocks of PI addresses, which cannot be aggregated, may not be reachable from all parts of the Internet. A larger block of PI addresses solves the problem.

If the customer should decide to change its service provider, renumbering of the network is not needed.



A multihomed customer can in some cases use PA addresses. The address space must be assigned from one of the ISPs. When the customer announces the block of PA addresses to both ISPs, both should propagate the addresses to the rest of the Internet. The provider that assigned the address space should also announce the larger block of addresses, of which the customer is announcing a subset. Other ISPs now receive two alternate explicit routes and an overlapping route summary. Filtering out explicit routes is more likely at this time because the other ISPs recognize these as routes that can be aggregated. If the other ISPs filter out the more explicit routes, the customer is still reachable as long as both providers are announcing the overlapping route summary.



Some customers decide to use private addresses within their network and do [NAT](#) at the connection point to the ISP. This setup means that customers require only a very small portion of public addresses from the ISP. In the figure, only customer [DMZ](#) has been assigned public addresses. The customer network is connected to the customer DMZ using two alternate firewalls with both firewalls doing NAT.

In this case, the customer requires only a very small block of public addresses. These addresses can be PA addresses. If the customer decides to change its service provider, addresses are renumbered only at the NAT point. The rest of the customer network does not need to be renumbered.

Customer AS Number Allocation

Depending on a customer connectivity to single service provider or to multiple service providers, the assigned [AS](#) number should be private or public.

Single-homed customers:

- If static routing is used, the AS number is not needed.
- If BGP is desired, private AS number (64512 - 65535) can be used.

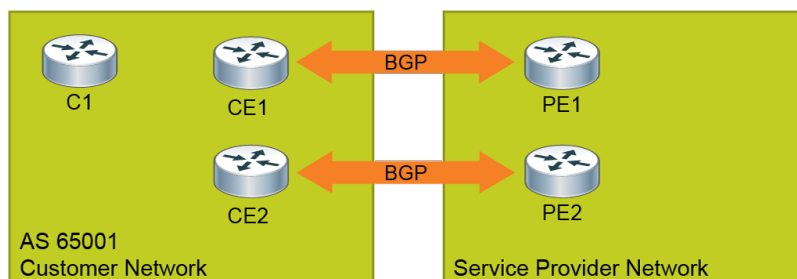
Multihomed customers:

- Public AS number has to be used.

Customers who are connected to a single service provider usually do not use [BGP](#), so an AS number is not needed. If a customer needs a BGP to detect link failures, the service provider can assign a private AS number to the customer.

Customers who are connected to multiple service providers use BGP and they should be assigned a public AS number.

AS Number Allocation: Single-Homed Customers



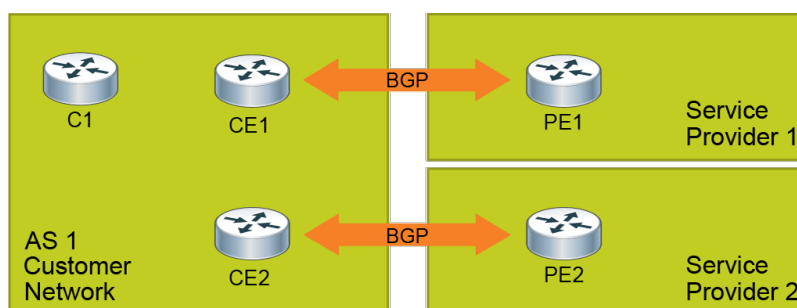
- Customers running BGP with the service provider need their own BGP AS number.
- **Private** AS numbers (64512-65535) can be used for customers who are connected to a single service provider.

BGP requires the use of AS numbers. When BGP is configured, the AS number is mandatory information. Public AS numbers are a scarce resource, however. Customers should use public AS numbers only when they are required. A customer that uses BGP to exchange routing information with only one [ISP](#) does not require a public AS number. This customer can use a private AS number.

An ISP network that is running BGP with some of its customers must determine whether a public or a private AS number is required for each customer. When the customer can use a private AS number, the ISP must allocate one from the range of private AS numbers (64512 to 65535). The ISP must make sure not to assign any of the private AS numbers to more than one customer.

When the ISP receives BGP routes from the customer, the ISP routers see the private AS number in the AS path. The ISP treats this private number as any other AS number. However, before the ISP propagates any of these routes to the rest of the Internet, it must remove the private AS numbers from the AS path. The same AS number may be in use by someone else. After the private AS number is removed, the route appears as belonging to the public AS of the ISP.

AS Number Allocation: Multihomed Customers



- Multihomed customers must run BGP with their service providers.

- Multihomed customers must use **public** AS numbers for their autonomous systems.

A multihomed customer requires a public AS number and must run BGP with both of its ISPs. The customer should not use a private AS number because both ISPs must propagate the customer routes to the rest of the Internet. If the customer does use a private AS number, and both ISPs remove the number before sending it to the rest of the Internet, then the customer routes will appear to be local in the public AS of both ISPs. To make BGP work correctly, multihomed customers need to avoid this situation.



With the help of the AS number translation feature, private AS numbers can also be used for multihomed customers, but this type of configuration is not encouraged.

Multihomed customers are correctly connected to the Internet by assigning a public AS number to the customer network. This public AS number appears in the AS path and service provider should propagate it to the rest of the Internet. The customer network is now reachable by the rest of the Internet through both providers. Internet endpoints use the route with the shortest AS path as the best route to the customer network.

Summary

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

- Different customers have different requirements, especially for redundancy for their Internet connections. These connectivity options include:
 - **Single-homed:** One or multiple connections to a single ISP.
 - **Multihomed:** Multiple connections to different ISPs.
- Routing, static or dynamic, that is used between a customer and service provider depends on the selected connectivity type.
 - If dynamic routing is required, BGP is the only option.
- IP addressing assignment depends on the selected connectivity type:
 - **Single-homed:** PA address space.
 - **Multihomed:** PI address space.
- AS numbering depends on the selected connectivity type:
 - **Single-homed:** No AS number, or private AS number
 - **Multihomed:** Public AS number.

Implementing Customer Connectivity Using Static Routing

Overview

When a customer can connect to the Internet through either a single connection to a service provider or multiple connections to the same [ISP](#), static routing is the simplest routing approach to implement between the customer and provider. When network administrators are implementing customer-to-provider connectivity with static routes, knowledge of static routing implementation guidelines will aid in successfully deploying static routing network configurations.

Upon completing this lesson, you will be able to:

- Identify when to use static routing between a customer and a service provider in a BGP environment
- Describe the characteristics of static routing between a customer and a service provider in a BGP environment
- Identify design considerations for propagating static routes in a service provider network
- Configure static route propagation in a BGP environment with different service levels
- Configure a typical backup setup that uses static routing between a customer and a service provider in a BGP environment
- Describe the limitations of floating static routes when they are used in typical backup static routing scenarios and the corrective actions to overcome these limitations
- Describe the characteristics of load sharing when you are configuring static routing between a customer and a service provider

When to Use Static Routing?

Static routing is the best solution to implement when you do not have redundancy in the network topology.

- Static routing should be used only with single-homed customers in the following examples:
 - Customers with a single connection to the Internet.
 - Customers with multiple connections to the same service provider in environments where link and equipment failure can be detected.
- Dynamic routing with BGP must be used with multihomed customers.

A single connection between the customer network and the service provider network does not provide any redundancy. If the link goes down, the connection is lost regardless of which routing protocol is configured in the customer or provider network. When there are redundant connections between the customer network and the network of a single service provider, static routing can be used under specific circumstances.

The [CE](#) router that is using an [IGP](#) must conditionally announce a static default route. If the link to one of the CE routers goes down, then the router must be able to detect the failure and invalidate the static default route. Announcement of this router as a default gateway that is using an IGP must now cease. Likewise, on the [PE](#) routers, the static routes that are pointing to the customer networks must be invalidated if the link between them goes down, and redistribution to [BGP](#) is therefore stopped.

If link-level procedures cannot detect a link failure, the interface remains in the up state. The static routes are not invalidated, and packets are forwarded into a "black hole." Since the router cannot detect a failure at the link level in these cases, BGP must be used between the customer and the provider.

BGP must also be used between the customer and the service provider networks when the customer is multihomed. This is true regardless of which link failure detection mechanisms are in use.

Characteristics of Static Routing

When static routing is implemented between the customer network and the [ISP](#) network, the [CE](#) must announce itself as a default gateway or a gateway of last resort. This procedure must be done using the [IGP](#) within the customer network. Different routers within the customer network must be able to select the best route to the exit point of the network.

- The customer network must announce a default route:
 - Redistribute default route into customer IGP if the customer is running EIGRP.
 - Use **default-information originate** if the customer is running OSPF or RIP.
- Customer-connected routes should be redistributed into service provider BGP.
 - Redistribute static routes into BGP, not core IGP.
- Specific customer routes should not be advertised to other autonomous systems:
 - Mark customer routes with the **no-export** community when redistributed into BGP.
 - Use static route tags for consistent tagging.

Different IGPs use different methods of announcing a router as a gateway of last resort. [EIGRP](#) uses the concept of a default network, while [OSPF](#) and [RIP](#) send reachability information about network 0.0.0.0/0. In either case, the network operators of the customer network are responsible for configuring their network to use the CE router as a gateway of last resort.

When static routing is used between the customer and the provider, the [PE](#) must propagate a static route that points to the customer network. It must propagate it to all other routers within the ISP network, and also to the rest of the Internet. The route is redistributed into [BGP](#).

Customer routes should not be redistributed into the IGP of the ISP network. Care should be taken that the IGP of the ISP network does not carry too many routes. Redistributing customer routes into the IGP could potentially cause poor performance and might eventually cause a complete shutdown of IGP routing at the service provider.

Imagine that a customer uses [PA addresses](#) and the ISP announces a large block of addresses for which the network of this customer is only a small portion of the block. The service provider should not propagate the routes of this customer to the rest of the Internet. The ISP should carefully redistribute only those connected routes that a single [IP address](#) customer uses. The rest of the Internet should receive only an announcement containing the larger block of addresses, a summary route.

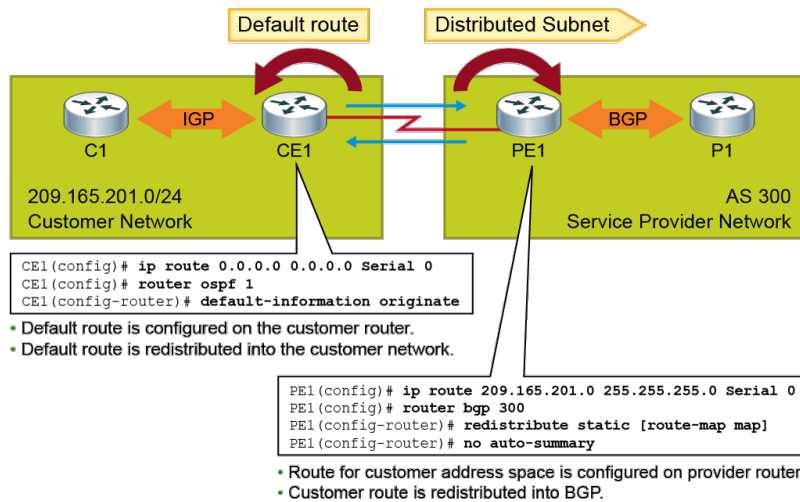
An easy way of achieving this setup is to use communities within the ISP network. You can make individual routes ineligible for advertising outside of the local [AS](#) by setting the predefined BGP **no-export** community. To ensure that the BGP communities are propagated, at least over all [IBGP](#) sessions, the network operators of the ISP network must configure a **send-community** option for all IBGP neighbors. If a router in a provider network receives an update carrying the **no-export** community, the router will not propagate the update to any external BGP neighbors. Meaning the update is not propagated to other service providers.



Communities are set by using route maps. A route map can select routes based on various attributes. One of these attributes is the route tag. Through configuration, the router can assign a route tag to specific static routes. This option means that the network operators of the ISP network can invent a scheme of tagging. For example all static routes that should not be propagated to other autonomous systems are assigned a specific tag. Then a route map can select all routes with that tag and assign them the **no-export** community.

Characteristics of Static Routing Example

In the figure, the customer network is connected to the Internet by using a single permanent connection to a single service provider.



In this case, a routing protocol does not add any redundancy and would only add complexity.

The configuration of CE and PE router is shown in the example. Access link is used, and directly connected routes are redistributed into BGP on the PE router. The CE router is configured with static default route, which points to the serial 0 interface. If the serial interface goes down, the route becomes invalid. The default route is also redistributed into OSPF, which is used in the customer network. Therefore the router announces a default route into OSPF as long as it has a valid default route itself.

The PE router also has a static route, declaring the customer IP network number 209.165.201.0/24 as reachable over the serial 0 interface. It also becomes invalid if the interface goes into the down state. The PE router must forward this information to all other ISP routers and to the rest of the Internet. This action is accomplished by redistributing the static route into BGP. As long as the static route is valid, the BGP announces it. To the rest of the Internet, the customer network appears as reachable within the AS of the ISP. As far as the rest of the Internet is concerned, the customer is a part of the service provider AS.

Designing Static Route Propagation in a Service Provider Network

You can easily extend the principle of using tags when you are configuring static routes. Based on those tags, you can assign different communities to implement a more complex routing policy.

- Identify all possible combination of services that are offered to a customer, including QoS services.
- Assign a tag to each combination of services.
- Configure a route map that matches the defined tags and sets BGP communities or other BGP attributes.
- Redistribute static routes into BGP through a route map.
- For each customer, configure a static route toward the customer with the proper tag.

To propagate static routes in a service provider network, complete these steps:

1. Identify all different service levels that are offered to customers and then all the different combinations of these service levels.
2. Assign each combination its own tag value and its own community.
3. Configure a route map, which selects routes with each of the assigned tags and sets the corresponding community value. Because the processing of a route map stops when the match clause of a statement is met, each route should be assigned a single combination of communities only. Therefore, you must take great care to assign a tag and a community combination to each combination of services that are provided.



When the [PE](#) routers redistribute static routes into [BGP](#), these routes must pass through the route map. BGP assigns the correct community depending on the tag values that are given on the configuration line for each of the static routes.

4. Finally, configure static routes. Before you configure a static route for a specific customer, you must identify the combination of the services that are provided to this customer. Then you must look up the corresponding tag value. After you have configured the route, you must assign the tag.

With this routing policy, every static route to a customer network is assigned a tag and the redistributed BGP route is assigned a corresponding community. The BGP communities that are attached to the routes signal to other routers in the [ISP](#) network which particular service combination you should use.

Static Route Propagation Example

A scenario with varied service levels is shown. Static route propagation is configured in a BGP environment.

Sample service offering:

- Addressing:
 - PA address blocks are not propagated to upstream ISPs.
 - PI address blocks are propagated to upstream ISP.
- Quality of service:
 - Normal customers.
 - Gold customers.
- Define static route tags.

Advertise Customer Route	QoS Type	Route Tag	Community Values
No	Normal	1000	no-export 300:31000
Yes	Normal	1001	300:31000
No	Gold	2000	no-export 300:32000
Yes	Gold	2001	300:32000

In this scenario, the service provider offers two different service levels to its customers: Normal and Gold. Customers are also assigned [IP address](#) blocks.

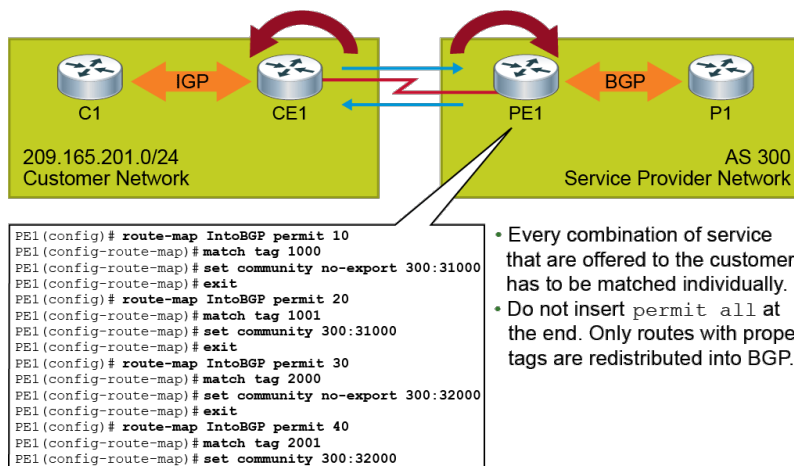
Some customers have [PA addresses](#), which the ISP does not announce as explicit routes. The large route summary block that the ISP announces covers these customers. Other customers use [PI addresses](#) that must be explicitly announced to the Internet by the service provider.

There are two different [QoS](#) services, Normal and Gold, and there are both PA and PI addresses. So the total number of combinations to cover the network policy is four:

- Normal QoS routes that the ISP assigns and should not be explicitly announced.
- Normal QoS routes that are [PI](#) routes and should be explicitly announced.
- Gold QoS routes that the ISP assigns and should not be explicitly announced.
- Gold QoS routes that are [PI](#) routes and should be explicitly announced.

Each of these four combinations receives its own tag value and its own community combination.

Configure Route Maps

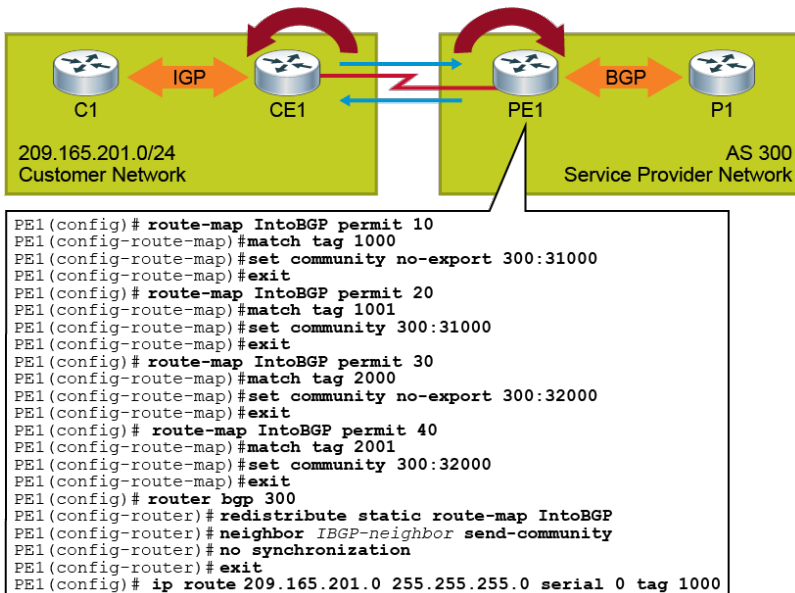


Network operators configure a route map in the PE1 router that has the static routes to the customer network. Redistribution of the configured customer static routes into BGP is also performed at the PE1 router.

A route map can match an individual route in a single route-map statement only. A single tag value, representing each combination of services must be assigned to the static routes by the router. When a route is matched, the interpretation of the route map for that individual route stops. The route map has one statement for each combination, and each statement matches a tag value and assigns the corresponding community combination for that tag.

The route map is applied during the redistribution of customer static routes into BGP at the PE router. Because the route map has no **"permit any"** statement at the end, the static routes that are not assigned any of the tags being used are not redistributed. The route map filters these routes out, forcing the network operators to make a tag assignment to all customer routes. Furthermore, the route map filtering can help the administrator catch configuration entry errors, thus giving all customers the service combination that they are entitled to.

Redistribution and Customer Routes



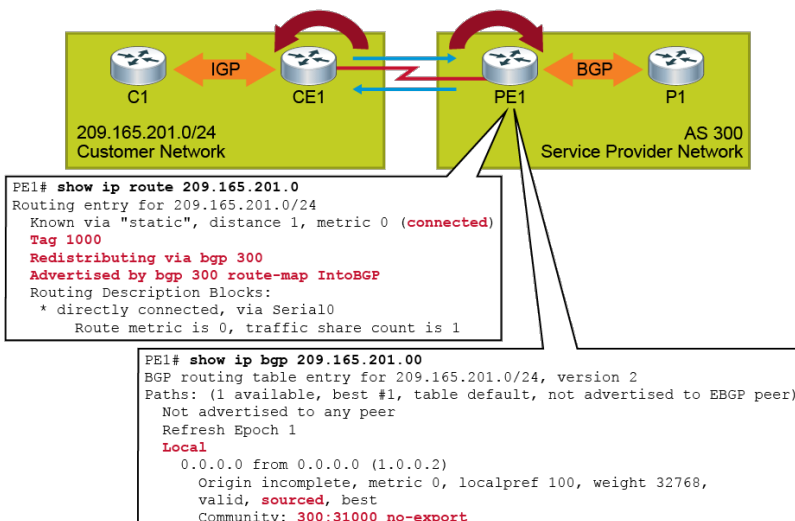
The figure shows how the PE1 router uses the route map named "IntoBGP" when redistributing static routes into BGP. Because the route map assigns community values that other routers within the ISP network will use, network operators must configure all **IBGP** neighbors with the **send-community** qualifier.

When connecting customers, the network operators identify which service combination to use for this particular customer. The three services that are associated with this particular customer are as follows:

- Apply normal QoS.
- Use a PA network number.
- Do not enable the provider to explicitly announce customer routes.

A static route to the customer is configured and assigned the appropriate tag value of 1000, which represents the specified services that are assigned to the customer.

Static Routes on the PE Router



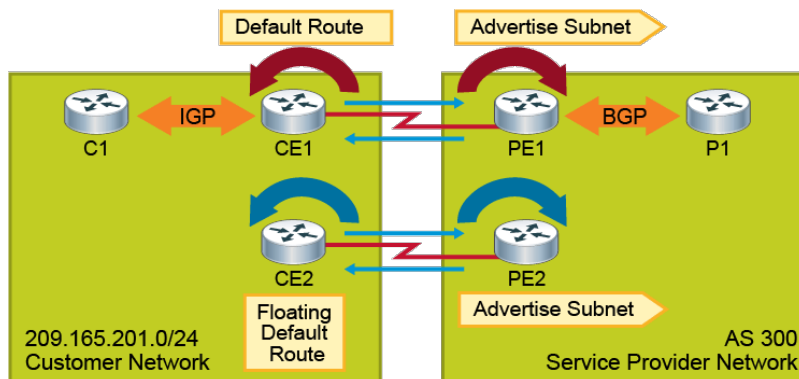
The **show ip route** command displays information from the routing table about subnet 209.165.201.0/24. The route is learned by static configuration and is redistributed via BGP. The router has assigned a tag value of 1000 to the customer route by using a statically assigned tag. The route must pass through the route map into BGP before being inserted into the BGP table.

The **show ip bgp** command displays information from the BGP table about subnet 209.165.201.0/24. The route is local within this AS and this router sourced it. The BGP communities 300:31000 and **no-export** have been assigned by the router to the redistributed customer route. They were assigned by using the provider-defined

route map before inserting the customer route into the BGP table.

BGP Backup with Static Routes

Imagine a case where the customer network has two connections to a single service provider, using two routers on each side. One connection between the customer network and the [ISP](#) is the primary connection, and the other connection is used for backup purposes only. If link-level procedures can detect link failures and a failure in the remote router, then static routing can be used between the customer and provider networks.



- Default routes on the CE router and redistribution into customer IGP:
 - Floating static routes are configured on the backup router.
- Static route on the PE router and redistribution into service provider BGP:
 - Floating static routes are configured on the backup router.
 - Separately tag primary and backup static routes and match them for redistribution.

As in the previous example, where no backup link is available, the CE1 router has a static default route toward the ISP. The PE1 router has static routes toward the customer. The customer router redistributes the static default route into its [IGP](#). The ISP router redistributes the static routes into [BGP](#).

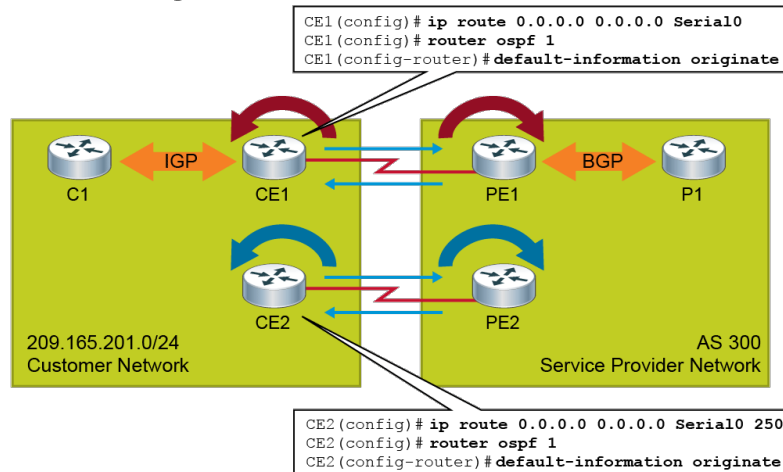
If the primary link goes down, the link-level procedures set the interface to the down state. This causes the static routes pointing out through the interface to be invalid and removing the routes from the routing table. When the interface changes back to the up state, the static route will reappear in the routing table.

The appearance of the route in the routing table conditions the redistribution of routes into any routing protocol. Thus, if the interface goes down, the router removes the static route from its routing table, and the route is withdrawn from the routing protocol. When the static route reappears, the redistribution process inserts it into the routing protocol again.

The CE2 router also uses a default static route toward the ISP via the backup link. The CE2 router is also redistributing the default route into the IGP. However, the static route that is used is a floating static route, which is assigned a high [AD](#)—higher than the administrative distance of the customer IGP. As long as the primary link works, the IGP provides the CE2 router with the primary default route. Because of the higher AD, the backup static default route is not installed into the CE2 routing table. Because the static route is not in the routing table, it is not redistributed. If the primary link fails, the IGP no longer feeds the CE2 router with a default route. The backup static default route is the only remaining default route. Therefore, the router will install the floating default route into its routing table and then redistribute it into the IGP.

The PE2 router can also use floating static routes, which are redistributed into the ISP [BGP](#) process.

Customer Configuration



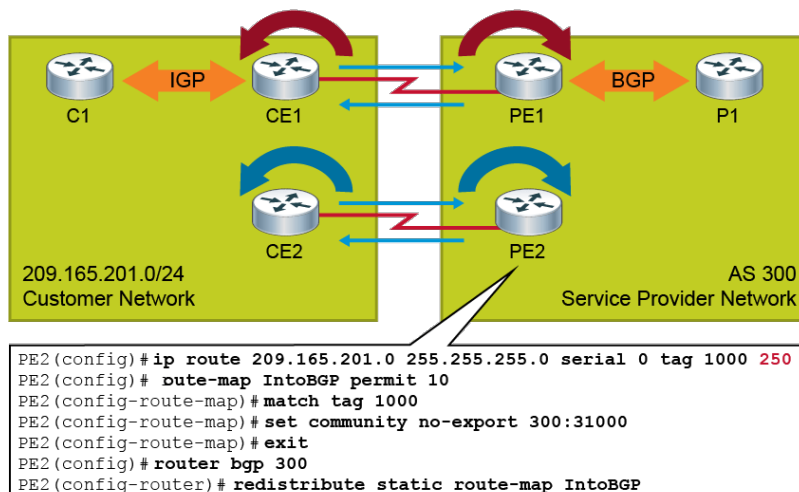
In the figure, the customer network and the ISP network are connected using leased lines. Both, CE1 and CE2, have a static default route toward the serial interface leading to the ISP. Both routers also do redistribution of the default route into the [OSPF](#) protocol, which is being used as an IGP within the customer network.

However, the static default route in the CE2 router is configured with an AD value set to 250. This AD value is higher than the AD values of any routing protocol. This configuration means that as long as the CE2 receives the default route by OSPF, the static default route is not used.

When the primary link goes down, the static default route in the primary router is not valid. The OSPF protocol stops announcing the default route, because the **default-information originate** command makes OSPF contingent on the availability of that static default route in the routing table before announcement.

The CE2 now installs its static default route in the routing table. The conditions for announcing the default route by OSPF are met and the rest of the customer routers see the CE2 as the gateway of last resort.

Provider Configuration



Caveat: The local BGP route is always better than an IBGP route. The floating static route is inserted into the BGP table and should not be removed there.

When floating static routes are configured on the [PE](#) routers, they are also redistributed into BGP. This configuration makes things a little bit more complicated.

The network operator configures a floating static route to the customer subnet 209.165.201.0/24. In the PE2 router, the floating static route is assigned the same tag value as the tag value being used in the PE1 router. The route map "IntoBGP" is the same as in the PE1 router and provides the routes to the customer network with the same communities. These communities are based on the same [QoS](#) level and the indication whether to explicitly announce route prefix to the rest of the Internet.

The floating static route has the administrative distance set to 250. This value is

higher than any routing protocol. When the PE2 router no longer receives any routing protocol information about the customer networks, the router will automatically install the floating static route and then redistribute it into BGP.

Based on BGP route selection rules, the redistributed floating static route will always remain the preferred path if an extra BGP configuration is not performed on the PE router. This preference means that regardless of whether the primary link comes back, the PE2 selects the locally sourced route as the best route. Therefore, the PE2 continues to announce a path toward the customer network. The backup link does not go back to the Idle state.

```
PE2# show ip bgp 209.165.201.0
BGP routing table entry for 209.165.201.0/24, version 7
Paths: (2 available, best #1, table default, not advertised to EBGP peer)
  Advertised to non peer-group peers:
    10.3.0.5
  Local
    0.0.0.0 from 0.0.0.0 (10.0.3.6)
      Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
      Community: 300:31000 no-export
  Local
    10.0.3.2 (metric 128) from 10.0.3.5 (1.0.0.2)
      Origin incomplete, metric 0, localpref 100, valid, internal
      Originator: 1.0.0.2, Cluster list: 10.0.3.5
      Community: 300:31000 no-export
```

- The BGP table on the service provider backup router contains the floating static route.

In this example, the **show ip bgp** command is used in the PE2 router of the provider to display the information about the customer network 209.165.201.0/24. The primary link has come back, so the PE2 now sees two alternate routes. The first route is the route that the router itself has redistributed into BGP using the floating static route. This route is locally sourced by this [AS](#) and has been assigned a weight value of 32768. The second route PE2 received by [IBGP](#) from the PE1 router. This AS also sources this route, but no weight value is assigned.

The BGP route selection algorithm selects the route with weight value 32768 as the best. As a result, the route that was received from the PE1 router is not a candidate to be installed in the routing table. It never competes with the floating static route. The floating static route stays in the routing table, and redistribution of the route continues until the backup link goes down and the route becomes invalid.

Floating Static Routes with BGP

Unfortunately, floating static routes do not work correctly with [BGP](#). After they are inserted, the floating static route is never removed from the routing table even if the primary link comes back.

Limitations and corrections:

- Floating static routes do not work correctly with BGP.
- Weight has to be lowered to default value for other BGP routes to be considered.
- BGP local preference has to be changed for floating static routes that are redistributed into BGP, to make sure that other routes take precedence.
- Administrative distance cannot be matched with a route map; extra tags need to be defined for static routes.

Whenever you use floating static routes in combination with redistribution into BGP, you will need to take extra configuration steps. You have to ensure that the BGP route selection algorithm selects the primary route as the best BGP route when it reappears:

- When a router redistributes a floating static route into BGP, the weight value that is assigned to the floating static route must be reduced (for example set to 0). Otherwise, the floating static route will always be selected as the best BGP route after the first failure of the primary link occurs.
- The router must also assign local preference values to the floating static route, so that the floating static route has a lower local preference than the primary route (for example 50). This assignment ensures that the primary route is selected as the best BGP route on other routers within Provider network.

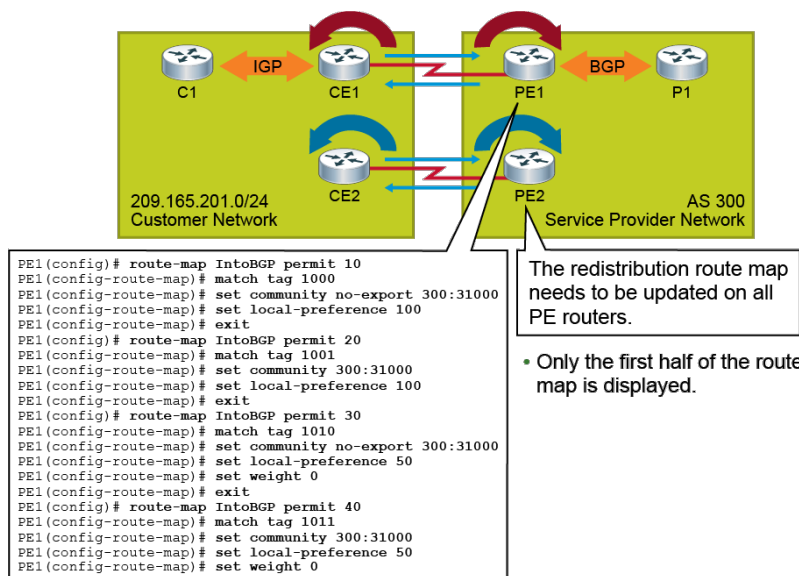
These two requirements must be specified on the [PE](#) router in the route map "IntoBGP" that is used for the redistribution. The route map must select the floating static routes and set weight and local preference. However, a route map cannot do matching based on the [AD](#) value that has been assigned to a static route. Some other means are required for the route map to distinguish between normal static routes with normal weight and local preference and the floating static ones with modified values.

The solution is to create extra tag values for this set of static routes. The tag value must not only reflect the [QoS](#) level and whether to announce the route. The tag value must also indicate if it is a primary route or a backup route.

Advertise Customer Route	Backup	QoS Type	Tag	Community Values	Local Preference
		Normal	1000	no-export 300:31000	100
	Yes	Normal	1010	no-export 300:31000	50
Yes		Normal	1001	300:31000	100
Yes	Yes	Normal	1011	300:31000	50
		Gold	2000	no-export 300:32000	100
	Yes	Gold	2010	no-export 300:32000	50
Yes		Gold	2001	300:32000	100
Yes	Yes	Gold	2011	300:32000	50

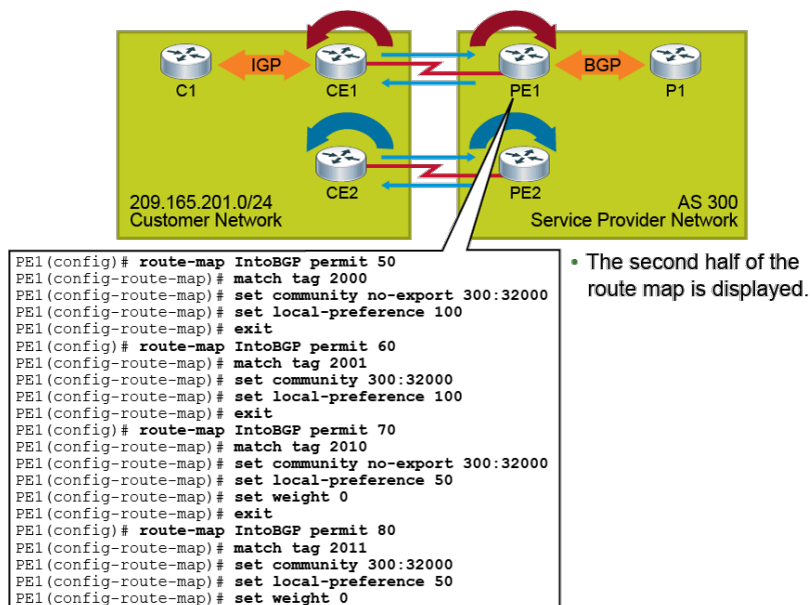
Eight tag values have currently been identified. Each tag value indicates a specific combination of explicit route propagation (backup or primary) and QoS level.

When network operators configure static routes in the PE router, they must consider which of the combinations that they should use for the route. The route map that they use when redistributing the static routes into BGP must be configured to recognize all eight combinations. It must set the appropriate weight and community and local preference values to these combinations.



The configuration output in the figure displays the first half of the "IntoBGP" route map. The output shows how match clauses identify four of the eight different tags. For each of the tag values, the route map sets the community, the local preference, and, in some cases, the weight.

The displayed half of the route map deals only with the four tags that indicate QoS Normal. So all statements in the configuration display have set the BGP community attribute to 300:31000.



The second part of the route map that is shown deals with the four tags that indicate QoS Gold, which are configured to set the BGP community attribute to 300:32000.

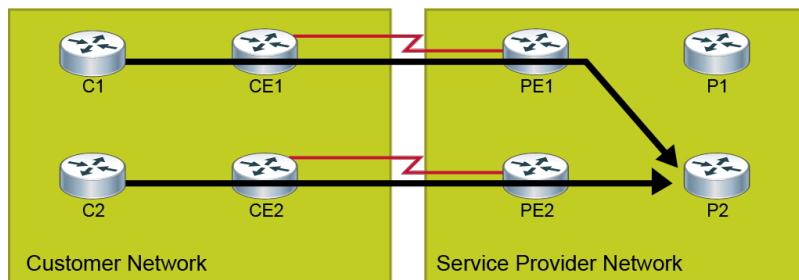
Tag values of 1000, 1010, 2000, and 2010 indicate that the route should not be explicitly propagated. The routes that a provider should not explicitly advertise to the rest of the Internet are assigned the **no-export** community by the route map.

Tag values 1010, 1011, 2010, and 2011 all indicate that the route is a backup route. Those tags have their weight value set to 0 and their local preference value set to 50. These settings ensure that on the return of a failed primary route, the PE router will select the primary route as its best path and remove the backup floating static route from its route table.

Load Sharing with Static Routes

A customer that has two connections to a service provider can also decide to balance traffic over both links. If both links are active, traffic should be balanced across both links. If there is link failure, one link should care for all traffic.

Outgoing Traffic Load Balancing



Outgoing traffic load balancing:

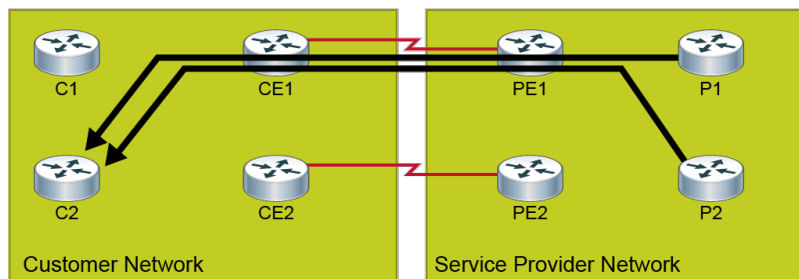
- On each CE router, the default route is redistributed into IGP.
- Each CE router uses the closest CE router as the exit point.
- Balanced load sharing is achieved if the CE routers are collocated.

Load sharing of outgoing customer traffic is accomplished by configuring a standard default static route in both [CE](#) routers. Each static route is valid as long as the serial link from the CE router to [PE](#) router is up. When both static routes are valid, both CE routers announce the default route into the customer network.

The remaining routers in the customer network see two candidate gateways of last resort. These remaining routers choose the closest one, with respect to the [IGP](#) metric. The part of the network that is closer to the uppermost exit point uses that exit point for all outgoing traffic. The other part of the network uses the other (lower) exit point.

If both exit points are collocated, they are equally distant from each of the other routers in the customer network. Each router within the customer network therefore uses load sharing of traffic sent out of both exit points.

Return Traffic Load Balancing



Load sharing of return traffic is impossible to achieve with multiple edge routers.

- All provider routers select the same BGP route to the destination.
- All return traffic arrives at the same PE router.

The service provider routers know about two routes to the customer network. Those routes are learned through [BGP](#). BGP was designed to select only one route, which is the best route from a collection of routes to destination, allowing no load sharing. The PE routers that receive the same BGP route from two CE routers will always select the closer CE router. If all other BGP attributes are equal, the [IBGP](#) route with the closer next hop is selected. The part of the [ISP](#) network that is closer to the uppermost connection uses that connection. The other part of the ISP network uses the other (lower) connection.

If both connection points are collocated, all PE routers select the same IBGP route based on router-ID (because the IGP metrics are always equal). So all the return traffic is sent over a single link toward the customer network, resulting in no load sharing.



Since Cisco IOS Software Release 12.2, the IBGP multipath load-sharing feature enables the BGP-speaking router to select multiple IBGP paths as the best paths to a destination. The best paths or multipaths are then installed in the IP routing table of the router.

You can optimize return traffic load sharing:

- Each PE router advertises only part of the customer address space into the provider backbone.
- Every PE router also advertises the whole customer address space for backup purposes.

Load sharing is not optimal—every link will carry return traffic for part of the customer address space.

To obtain better control of the return traffic load, the customer address space must be advertised to the PE routers using multiple, more explicit routes. The upper edge router could advertise half the address space, and the lower edge router could advertise the other half. For backup reasons, they also should both advertise the entire address space as a larger route summary.

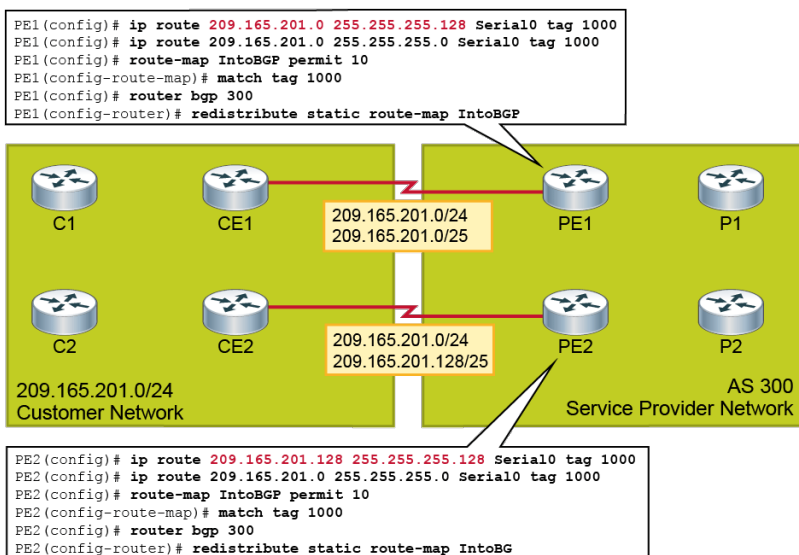
As long as both paths are available, the traffic from the ISP to the customer uses the most explicit route. In this case, two explicit routes are used to send the traffic representing one half of the address space over one link and traffic representing the other half of the address space over the other link.

Load sharing in this way does not result in an equal load on the links but rather a statistically based distribution of the traffic load over the links.

Load Sharing with Static Routes Example

You can achieve load balancing of the incoming traffic by advertising different parts of the customer network by different PE routers. The upper PE router could advertise one-half of the address space, and the lower PE router could advertise the other half. For backup reasons, the PE routers also should advertise the entire address space as a larger route summary.

In the example, the customer address space 209.165.201.0/24 is partitioned into two smaller blocks: 209.165.201.0/25 and 209.165.201.128/25.



The PE1 router advertises the route to 209.165.201.0/25, and the PE2 router advertises the route to 209.165.201.128/25. Both PE routers also advertise the entire address space 209.165.201.0/24.

As long as both paths are available, the traffic from service provider to the customer uses the most explicit route. Two explicit routes are used to send traffic. Traffic to destinations in the 209.165.201.0/25 range is sent to the upper connections point. Traffic to destination in the 209.165.201.128/25 range is directed to the lower connection point.



Load sharing in this way does not result in an equal load on the links but rather a statistically based distribution of the traffic load over the links.

Summary

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

- Static routing can be used for:
 - Customers with single connection to service provider.
 - Customers with multiple connections to service provider—primary and backup path, or load balancing.
- In static routing, the customer network must announce a default route.
- Customer routes should be carried in BGP, not core IGP; and routes to subnets of the provider address block should not be propagated to other ASs.
- When you are using static routes in a backup scenario, floating static routes are used on the backup routers.
- Depending on the origin of the customer address space, the provider may elect not to advertise the customer space, choosing to advertise a larger aggregate route instead.

Connecting a Customer to a Single Service Provider

Overview

When multiple connections to the same service provider are the only means that a customer has of connecting to the Internet, it is important that the connections are correctly configured. They must ensure proper interaction between the customer and the service provider network. It is also important to understand how to configure routing protocols so that customer backup or load-balancing requirements are met.

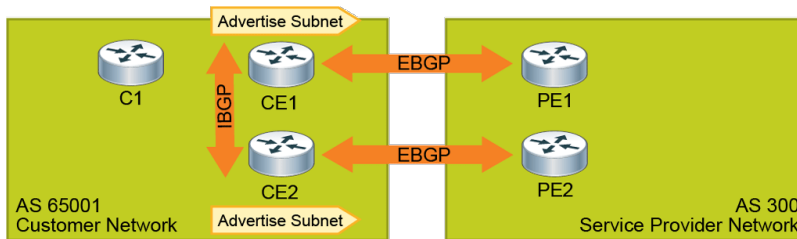
In this lesson, you will learn about the use of multiple connections between a customer and a single [ISP](#) for backup and load-sharing purposes.

Upon completing this lesson, you will be able to:

- Configure BGP on a customer network to establish routing between a multihomed customer and a single service provider
- Configure conditional advertising of a customer address space when you are using BGP to establish routing between a multihomed customer and a single service provider
- Configure BGP on a service provider network to establish routing between a multihomed customer and a single service provider
- Disable the propagation of private AS numbers to EBGp peers in a service provider network where a multihomed customer is advertising private numbers in the AS path
- Describe the BGP Support for Dual-AS Configuration for Network AS Migrations feature
- Configure a typical backup setup between a multihomed customer and a single service provider in a BGP environment
- Describe how you can implement load sharing between a multihomed customer and a single service provider in a BGP environment
- Identify the Cisco IOS command that is required to configure load sharing between a multihomed customer and a single service provider using BGP multipath
- Configure load sharing between a multihomed customer and a single service provider using EBGp multihop

BGP Configuration on Customer Routers

In the example, the customer network is connected to a service provider network by using multiple permanent links. BGP is used to exchange routing information between the customer and the provider.



- BGP is run between the customer and the service provider.
- The customer advertises allocated address space into BGP. The customer can use a private AS number.
- The service provider advertises only the default route to the customer. CE routers advertise the default route to the customer network by using IGP.
- The service provider has to deploy inbound BGP filters.

Selecting BGP as the routing protocol between the customer and the provider network ensures that a link failure or the failure of a remote router is detected. In this scenario, the customer does not require the use of a public AS number or full Internet routing. Instead, a private AS number is assigned to the customer network, and the ISP sends a default route to the customer through BGP.

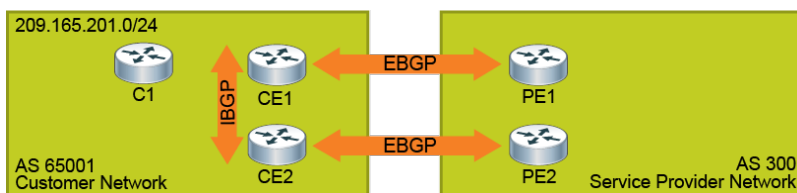
There is a significant difference in this case, as compared to a network scenario where static routes and redistribution are used. Routers within the private AS of the customer now advertise the customer routes via BGP. Thus, the customer is responsible for announcing its own address space. The ISP receives routes from the customer and conditionally propagates them (similar to static routing). If the customer uses PA address space and the ISP can summarize the address space, it will not propagate the explicit routes from the customer to the Internet. The private AS number in the AS path attribute must be removed before the ISP can propagate any of the customer routes.

The customer is now creating BGP routes that the ISP receives. So any error that the customer makes can influence routing operation within the provider network and if propagated, within the Internet as a whole. Announcing a route to a network to which the customer has not been assigned may cause routing problems. There is always a risk that such routing problems can occur in a service provider network. However, the risk is much greater when the customer, whose network administrators usually have less experience with BGP, enters the configuration.

To reduce the risk of erroneous route advertising, the ISP should always filter any BGP information that comes from the customer network. The ISP should reject routes to networks that are not expected to be in the customer AS. Routes that contain an AS path with unexpected AS numbers should also be rejected.

BGP Configuration on Multihomed Customer Routers Example

In the figure, the customer has been assigned the private AS number 65001.



```
CE1(config)# ip route 209.165.201.0 255.255.255.0 null 0
CE1(config)# router bgp 65001
CE1(config-router)# neighbor 209.165.201.2 remote-as 65001
CE1(config-router)# neighbor 172.16.33.33 remote-as 300
CE1(config-router)# network 209.165.201.0 mask 255.255.255.0
CE1(config-router)# exit
CE1(config)# router ospf 1
```

```
CE1(config-router)# default-information originate
```

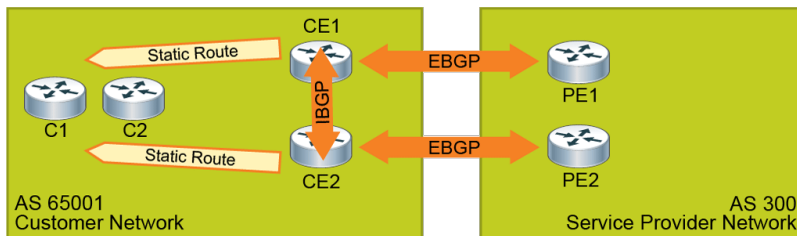
- The customer address space is advertised on every CE router.
- CE routers run IBGP between themselves and advertise the default route to the rest of the customer network.

Both [CE](#) routers are configured to run BGP and should advertise all of the customer networks with the **network** command. If only one router advertises the network, a single point of failure has been introduced. The two customer edge routers must also run [IBGP](#) between them to make common decisions regarding BGP routing information.

Each CE router has an [EBGP](#) session with the [PE](#) router on the other side of the link. Over that EBGP session, the ISP announces only a default route to the customer AS. When EBGP receives the default route, it installs it in the routing table and redistributes it into the [IGP](#), in this case [OSPF](#), of the customer.

Conditional BGP Advertising in Customer Networks

The customer should announce addresses as large as possible (the larger the address space that can be aggregated, the better). The BGP advertisement is configured on the CE routers using the **network** command. The appearance of a corresponding network or subnet in the routing table of the edge router conditions the route advertisement. Because there is a chance that the customer subnetted the address space into smaller subnets, the complete customer address space will not be presented in the CE routing table. The solution is to manually enter the route into the routing table by configuring a static route to null 0. However, the static route is always present in the routing table. And the BGP advertisement is always performed from the CE router, even if the CE router cannot access the rest of the customer network.



- The CE router should announce the customer address space into BGP.
- The CE router should advertise the customer space only if the customer network is reachable from the CE router (**conditional advertising**).
- The CE router should stop advertising the customer address space if the CE router loses connectivity with the customer core.
- The reachability of the customer core network is tested using a static route. The static route should point to the core network next hop that is learned via IGP, not to null 0.

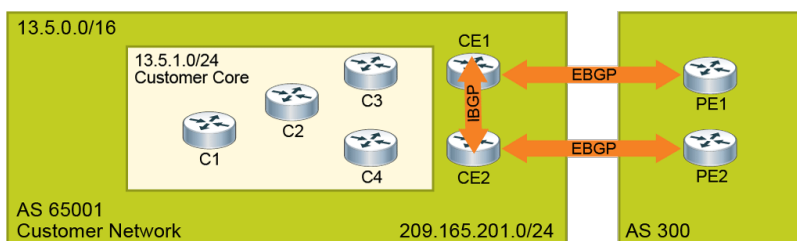
If the CE router loses connectivity to the rest of the customer network but is still connected to the ISP network, the BGP advertisement must cease. In this case, BGP advertisement can be stopped if BGP advertisements are bound to the reachability status of a specific subnet in the core of the customer network, according to the customer IGP.

The problem with using a static route to null 0 is that it conditions the network statement in the BGP configuration so that BGP always advertises the route. If the CE router loses connectivity with the rest of the customer network, the router continues to advertise the entire customer address space. The ISP network receives a valid route from the CE router. Traffic is sent to this router, but because the router has lost connectivity with the rest of the network, the traffic is dropped. Or the traffic is routed to the null 0 interface using the static route.

The solution is to create a static route for the customer-assigned addresses, which points to a next hop that was learned through an IGP. The static route will be in the routing table only if the next hop is reachable. The static route would be valid and the route would be advertised through BGP. If the CE router loses connectivity from the rest of the customer network, the next hop becomes unavailable and the static route is not valid anymore. Because the customer route is not present in the routing table anymore, BGP will cease to advertise the customer route to the ISP. The customer network will then be accessible over the backup connection, if available.

Conditional BGP Advertising in Multihomed Customer Networks Example

In this example, the customer network also uses the address space 13.5.0.0/16.



```
CE2(config)# ip route 13.5.0.0 255.255.0.0 13.5.1.1
CE2(config)# router bgp 65001
CE2(config-router)# neighbor 209.165.201.1 remote-as 65001
CE2(config-router)# neighbor 172.16.33.33 remote-as 300
CE2(config-router)# network 13.5.0.0 mask 255.255.0.0
CE2(config-router)# exit
CE2(config)# router ospf 1
CE2(config-router)# default-information originate
```

The address space is further subnetted at the customer site. One of the subnets (subnet 13.5.1.0/24) is identified as being the central part of the customer core network.

The CE routers participate in the IGP routing of the customer. This participation means that these routers have information about which subnets within the address space 13.5.0.0/16 are currently reachable. If these subnets are available, there is an explicit route to each of them. If any of the subnets go down, or if the path toward them goes down, the route to that subnet is removed from the routing table.

The BGP advertisement in each of the CE routers is configured to advertise the full address space that the customer uses. When the CE routers advertise this route, the ISP network, and the rest of the Internet, see customer's complete address space as one single route, 13.5.0.0/16.

The appearance of the static route, IP route 13.5.0.0 255.255.0.0 13.5.1.1 conditions the advertisement of the customer address space by BGP. If the static route is valid, then the BGP route 13.5.0.0/16 is advertised. The static route is a recursive route—the router takes another look in the routing table for the address 13.5.1.1 before determining what to do with the static route. The idea is that 13.5.1.1 is reachable via the IGP. The IGP announces the subnet 13.5.1.0/24. If this subnet is reachable by the edge router, then the static route to 13.5.0.0/16 is valid. If there is no route to 13.5.1.1, then the static route is invalid.

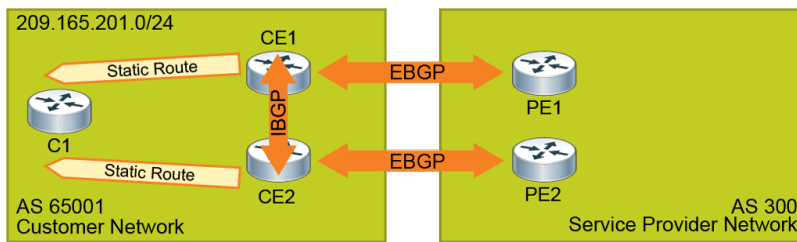


The condition, whether or not to advertise the entire customer address space 13.5.0.0/16, is controlled by the IGP reachability of a single subnet, 13.5.1.0/24.

The IGP configuration also includes origination of the default route by both CE routers.

BGP Configuration on Service Provider Routers

In the [ISP](#) network, the two [PE](#) routers must have [BGP](#) sessions configured with the customer. There is no point in feeding the full Internet routing table to the customer. The table contains the same set of routes for both links and the customer always uses the ISP for all traffic toward the Internet. Injection of a default route in the customer network would accomplish the same task.



The service provider must:

- (Conditionally) Advertise a default route to the customer through BGP.
- Filter incoming BGP updates with a prefix list to verify that the customer announces only the assigned address space
- Filter incoming BGP updates with an AS path filter list to verify that the customer uses only its own AS number
- Optionally, the **no-export** community should be set on customer routes.

The customer is responsible for its own advertisements. Because customers are much less likely to be experienced in BGP configuration than the ISP, they are more likely to make errors. Therefore, the ISP must protect itself and the rest of the Internet from those errors.

The service provider should use a filter that allows only customer-assigned routes and denies any other route. This filter ensures that private address space or any other illegal networks that the customer erroneously announces, never reach the ISP BGP table. Filtering based on the [AS](#) path also provides some protection from customer configuration errors. Only routes that originated within the customer AS should be allowed. A filter list performs this check.

If the customer address space is [PA address](#) space and it represents only a small part of a larger block that the ISP announces, the explicit BGP routes that are received from the customer do not need to be advertised to the rest of the Internet. The ISP can announce the large block, attracting any traffic toward any subnet within the block. After the traffic enters the ISP network, the more explicit routes to the customer network are available and used. In this case, the [PE](#) router can tag the BGP routes that are received from the customer with the **no-export** well-known community. This community prevents the ISP from sending the routes to any other AS.

```
router(config-router)# neighbor ip-address default-originate
```

- By default, the default route (0.0.0.0/0) is not advertised in outgoing BGP updates.
- The **neighbor default-originate** command advertises the default route to a BGP neighbor even if the default route is not present in the BGP table.



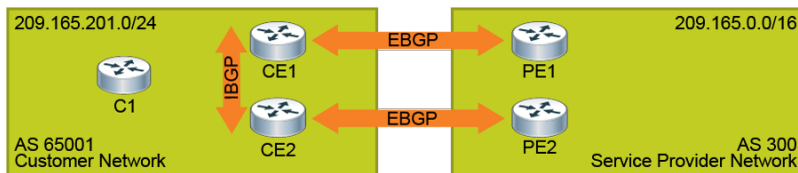
The default route is not sent through the outbound BGP filters (prefix list, filter list, or route map).

The default route, 0.0.0.0/0, is not advertised in outgoing BGP updates unless it is explicitly configured. The **neighbor default-originate** router configuration command is used to initiate the advertisement of the default route to a neighbor.

BGP does no checking before the default route is advertised. The default route does not need to be present in the BGP table before it is advertised using this command. The default route is also sent without being filtered by any outgoing prefix lists, filter-lists, or route maps.

BGP Configuration on Service Provider Routers Example

This example shows the configuration of the PE router.



```
PE1(config)# router bgp 300
PE1(config-router)# neighbor 172.16.33.3 remote-as 65001
PE1(config-router)# neighbor 172.16.33.3 default-originate
PE1(config-router)# neighbor 172.16.33.3 prefix-list DefaultOnly out
PE1(config-router)# neighbor 172.16.33.3 prefix-list Customer1 in
PE1(config-router)# neighbor 172.16.33.3 filter-list 15 in
PE1(config-router)# neighbor 172.16.33.3 route-map AllCustomerIn in
PE1(config-router)# exit
PE1(config)# ip as-path access-list 15 permit ^65001(_65001)*$
PE1(config)# ip prefix-list DefaultOnly permit 0.0.0.0/0
PE1(config)# ip prefix-list Customer1 permit 209.165.201.0/24 le 32
PE1(config)# ip prefix-list Provider permit 209.165.0.0/16 le 32
PE1(config)# route-map AllCustomersIn permit 10
PE1(config-route-map)# match ip prefix-list Provider
PE1(config-route-map)# set community no-export additive
PE1(config-route-map)# exit
PE1(config)# route-map AllCustomersIn permit 1000
```

The customer is assigned the private AS number 65001. The BGP session is opened with the customer [IP address](#), 172.16.33.3.

The ISP sends the default route only to the customer. This route is configured using first the **default-originate** command and then the prefix list DefaultOnly.

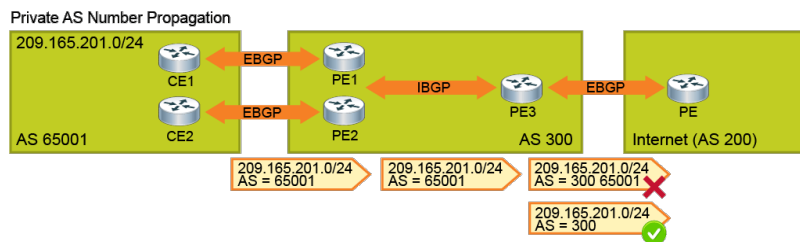
Received routes from the customer must first pass the prefix list Customer1. There is one dedicated prefix list for each individual customer; the prefix list permits only the routes that the customer is allowed to announce. If the prefix list allows the routes, the routes must also pass the filter list named "15." In this case, the filter allows the private AS of the customer in any number of repetitions, as long as it is the only AS number in the path. This filter list allows AS path prepending configurations on the customer side. If both prefix lists and the filter list allow the received route, then the route map AllCustomersIn is applied.

The route map is a general route map that is used for all customers. It checks every route that is received, via the prefix list Provider. If the route is within the big block of PA address space that the ISP announces to the rest of the Internet, the customer route is marked with the **no-export** community. This mark means that the route is used within the ISP AS only and is not sent to the rest of the Internet.

Routes that are received from the customer, and are allowed by the prefix list and filter list, but do not fall within the PA address space, are allowed by the route map and are not changed in any way. The ISP propagates these routes to the rest of the Internet.

Removing a Private AS Numbers

Routes that the ISP receives from the customer are propagated to the rest of the Internet only if they are part of the [PI address](#) space.



A customer can use a private AS.

- Private AS numbers should not be advertised into the Internet.
- Private AS numbers must be removed from the AS path before the customer BGP routes are advertised to other service providers.
- A private AS number can be removed from the tail of the AS path using the **remove-private-as** keyword.

When the ISP receives [BGP](#) routes from the customer, the [AS](#) path attribute of the received routes contains only the AS number of the customer. If the customer uses AS path prepending, there may be several repetitions of the customer AS number in the AS path. If the service provider propagates the customer routes to the Internet, the AS number of the customer will be present in the AS path unless it is explicitly removed.

i If the customer has been assigned a private AS number, this AS number must never be advertised by any router to the rest of the Internet. The service provider has to make sure that the private AS number is removed from the BGP route and that the service provider AS number is present in the tail of the AS path.

Removal of a private AS number from the AS path is accomplished by using the **remove-private-as** command on the ISP [EBGP](#) sessions with the rest of the Internet. In the figure, removal of the private AS number occurs on the EBGP session between AS 300 and the Internet.

```
router(config-router)# neighbor ip-address remove-private-as
```

- The command modifies AS path processing on outgoing updates that are sent to the specified neighbor.
- Private AS numbers are removed from the tail of the AS path before the update is sent.
- Private AS numbers followed by a public AS number are not removed.
- The AS number of the sender is prepended to the AS path after this operation.

To remove private AS numbers from the AS path in outbound routing updates, use the **neighbor remove-private-as** router configuration command. To disable this function, use the **no** form of this command.

```
neighbor {ip-address | peer-group-name} remove-private-as
```

Syntax Description

Parameter	Description
<i>ip-address</i>	IP address of the BGP-speaking neighbor
<i>peer-group-name</i>	Name of a BGP peer group

Use this command on the service provider egress routers. Before the service provider advertises any of the customer routes of the ISP to the rest of the Internet, the private AS numbers must be removed. The command removes those AS numbers if they are at the tail end of the AS path.

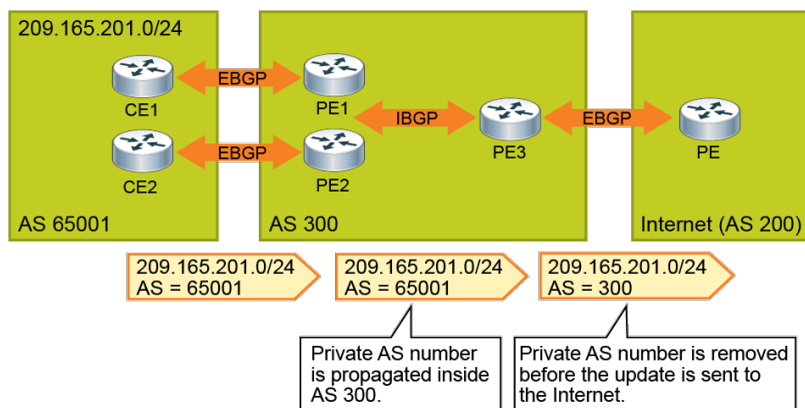


Private AS numbers followed by public AS numbers are not removed because the command's visibility is only on the last (tail end) AS number.

The AS number of the ISP is automatically prepended to the AS path attribute after the **remove-private-as** operation has completed. This situation means that the AS number of the ISP has not already been prepended to the AS path attribute when the tail of the AS path is checked for private AS numbers.

Removing a Private AS Numbers Example

In this example, the service provider AS (300) receives routes from the customer.



```
PE3(config)# router bgp 300
PE3(config-router)# neighbor 10.2.3.3 remote-as 200
PE3(config-router)# neighbor 10.2.3.3 remove-private-as
```

The ISP assigns the private AS number 65001 to the customer; therefore, the routes that the provider receives have an AS path containing only AS 65001. This information should be kept and used within the ISP network and should never be propagated to the rest of the Internet (AS 200 in this example).

The PE3 router in AS 300 has been configured to remove private AS numbers on EBGP routes toward AS 200. If private AS numbers appear in the tail end of the AS path (before AS 300 is added), they are removed.

This configuration must be applied to all egress routers in AS 300 that serve EBGP neighbors leading to other ISPs. No private AS number may be present in an AS path of a route that is propagated to a network using a public AS number.

BGP Support for Dual AS Configuration for Network AS Migrations

AS migration can be necessary when a telecommunications provider or [ISP](#) purchases another network. It is desirable for the provider to be able to integrate the second AS without disrupting existing customer peering arrangements. However, the amount of configuration that is required in the customer networks can make this migration a cumbersome task that is difficult to complete without disrupting service.

- Allows you to merge a secondary AS under the primary AS without disrupting customer peering sessions.
- Allows a router to appear to external peers as a member of secondary AS during the AS migration.
- Allows a network operator to merge the ASs and then later migrate customers to new configurations during normal service windows without disrupting existing peering arrangements
- If misconfigured, increases the possibility of creating routing loops

The [BGP](#) Support for Dual AS Configuration for Network AS Migrations feature allows you to merge a secondary AS under a primary AS without disrupting customer peering sessions. The configuration of this feature is transparent to customer networks. This feature allows a router to appear to external peers as a member of secondary AS during the AS migration. It also allows the network operator to merge the AS and then later migrate customers to new configurations during normal service windows without disrupting existing peering arrangements.

The **neighbor local-as** command is used to customize the AS path attribute by adding and removing AS numbers for routes that are received from [EBGP](#) neighbors. Use this command in address-family or router configuration mode. To disable AS path attribute customization, use the no form of this command.

```
neighbor ip-address local-as [as-number[no-prepend [replace-as [dual-as]]]]
```

Syntax Description

Parameter	Description
<i>ip-address</i>	Specifies the IP address of the EBGp neighbor.
<i>as-number</i>	Specifies an AS number to prepend to the AS path attribute. The range of values for this argument is any valid AS number from 1 to 65535.
no-prepend	(Optional) Does not prepend the local AS number to any routes received from the EBGp neighbor.
replace-as	(Optional) Prepends only the local AS number to the AS path attribute. The AS number from the local BGP routing process is not prepended.
dual-as	(Optional) Configures the EBGp neighbor to establish a peering session using the real AS number from the local BGP routing process. Or by using the AS number that is configured with the ip-address argument (local-as)



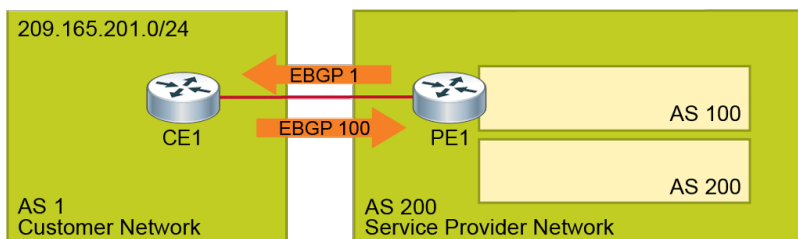
AS path customization increases the possibility that routing loops can be created if it is misconfigured. The larger the number of customer peerings, the greater the risk. You can minimize this possibility by applying policies on the ingress interfaces to block the AS number that is in transition or routes that have no local-as configuration.



BGP prepends the AS number from each BGP network that a route traverses to maintain network reachability information and to prevent routing loops. This feature should be configured only for the AS migration and should be deconfigured after the transition has been completed. This procedure should be attempted only by an experienced network operator, because routing loops can be created with improper configuration.

Dual-AS Configuration Example

The following examples show how Dual AS Configuration for Network AS Migrations feature is used to merge two autonomous systems without interrupting peering arrangements with the customer network.



- If a service provider migrates to a new AS number, customers have to change the BGP configuration.
- A merged service provider can present itself to the customer using the old AS number, so the customer is not required to change the BGP configuration.

The **neighbor local-as** command is configured to allow PE1 to maintain peering sessions through AS 100 and AS 200. CE1 is a customer router that runs a BGP routing process in AS 1 and is configured to peer with AS 200.

```
PE1(config)# router bgp 100
PE1(config-router)# no synchronization
PE1(config-router)# neighbor 172.16.33.3 remote-as 1
PE1(config-router)# neighbor 172.16.33.3 local-as 200 no-prepend replace-
as dual-as
```

```
PE1(config)# router bgp 200
PE1(config-router)# neighbor 172.16.33.3 remote-as 1
```

```
CE1(config)# router bgp 1
CE1(config-router)# neighbor 172.16.33.34 remote-as 200
```

After the transition is complete, the configuration on CE1 can be updated to peer with AS 100 during a normal maintenance window or during other scheduled downtime.

```
CE1(config-router)# neighbor 172.16.33.34 remote-as 100
```

Dual-AS Confederation Configuration Example

The following example can be used in place of the PE1 configuration in the previous example. The only difference between these configurations is that in this example PE1 is configured to be part of a confederation.

```
PE1(config)# router bgp 65100
PE1(config-router)# no synchronization
PE1(config-router)# bgp confederation identifier 100
PE1(config-router)# neighbor 172.16.33.3 remote-as 1
PE1(config-router)# neighbor 172.16.33.3 local-as 200 no-prepend replace-
as dual-as
```

Replace-AS Configuration Example

The following example strips private AS 65100 from outbound routing updates for the 172.16.33.3 neighbor and replaces it with public AS 1.

```
PE1(config)# router bgp 65100
PE1(config-router)# neighbor 172.16.33.3 local-as 1 no-prepend replace-as
```

Backup Solutions with BGP

When a customer uses [BGP](#) on multiple links between its network and the [ISP](#) network, the customer is solely responsible for controlling how it uses the links. The customer can choose to use its links in a primary or backup scenario or in a load-sharing scenario.

The CE routers control the route selection entirely.

- Outgoing traffic:
 - Local preference is used to select primary or backup link.
- Incoming traffic:
 - MED is used to advertise the primary or backup link to the ISP. This is not reliable, because the ISP can change the local preference.
- No service provider configuration is required.

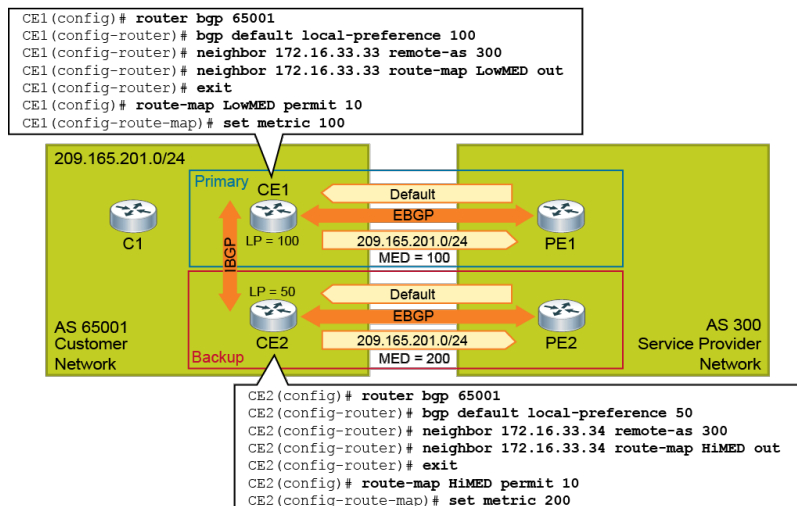
If one link is primary, then the other should be used for backup only. The customer can use the local preference configuration to direct all outgoing traffic over the primary link.

Incoming traffic to the customer is controlled by using either [AS](#) path prepending or the [MED](#). Because the customer has multiple connections to the same AS, the MED is the ideal attribute to use. When the customer announces its routes to the ISP, a worse (or high) MED value on the backup link is set. Also a good (or low) value on the primary link is set.

The receiving [EBGP](#) peer checks MED and AS path length only if the weight and local preference attributes have not been configured. In this case, the ISP should not use any of these configuration options. The ISP should rely solely on the attributes that it has received from the customer. However, if the ISP sets, for example, the local preference for customer routes, the ISP will control the path selection for traffic to the customer. This path selection is why you cannot rely on the fact that your traffic distribution scenario will work as intended.

Primary/Backup Link Selection Example

The customer is connected to the ISP over two permanent connections. The customer uses the upper connection as the primary connection and the lower connection as the backup.



The BGP configuration on the ISP side is transparent. This transparency means that no particular preference is configured to use the upper or lower connection. The ISP relies on the attribute values that are received from the customer.

The CE1 router is configured to set local preference to the value 100 on all EBGP routes that are received. The CE2 router sets the local preference attribute to a value of 50. This configuration means that the outgoing traffic toward any destination that the ISP announces is primarily sent over the upper link. It makes the primary router the preferred one.

Incoming traffic to the customer is directed to the primary link by using the MED. On the CE1 router, all routes that are sent to the ISP have their MED attribute set to the value 100 by the route map named "LowMED out." On the CE2 router, all routes that

are sent to the ISP have their MED attribute set to the value 200 by the route map named "HiMED out." Because the ISP receives the routes with all other attributes set to the same values, the MED values direct traffic for the customer to the primary link. This makes the primary router the preferred one from the service provider point of view.

Load Sharing

Load sharing of outgoing traffic from the customer network is identical to the static routing scenario. The customer [IGP](#) is configured to send information about a gateway of last resort. There is no difference whether the edge router gets its default by static routing or by incoming [EBGP](#) updates.

You can implement load sharing of return traffic in a number of ways:

- Outgoing traffic:
 - Load sharing is identical to the static routing scenario. The IGP determines it.
- Incoming traffic:
 - Announce portions of the customer address space to each upstream router.
 - Configure BGP multipath support in the service provider network.
 - Use EBGP multihop in environments where parallel links run between a pair of routers.

Load sharing of the return traffic coming back to the customer network from the [ISP](#) can be implemented in a number of ways:

- The customer can divide its address space into several announcements. The [CE](#) router can send each announcement over one of its EBGP sessions with the ISP. For backup purposes, the customer should advertise the entire address space over all of its EBGP sessions. The ISP now uses the most explicit route rule. As long as both links are up, traffic with destinations within one part of the customer address space is routed over one of the links. Traffic to the other part is routed over the other link.
- If the customer announces equivalent routes over both links, the ISP routers use the closest connection regarding the IGP of the ISP. If an ISP router has an equivalent distance to both connection points, the use of the **maximum-paths** (or [BGP](#) multipath) option causes load sharing.
- If the multiple links between the customer and the ISP network terminate in one single router on the customer side and one single router on the ISP side, the two routers must establish their EBGP session from loopback interface to loopback interface. Static or dynamic routing is required for one router to get information on how to reach the loopback interface of the other router. The use of the **ebgp-multihop** option is also required because the address of the neighbor is not directly connected.

Load Sharing with BGP Multipath

By default, [BGP](#) route selection rules select one, and only one, route as the best. If there are two identical routes, the tiebreaker is either the most stable route or the router-ID of the peer router that is advertising the route. However, when the **maximum-paths** router configuration command is used, the BGP route selection process will select more than one route as best if they are identical. The routes are all installed in the routing table, and load sharing takes place.

BGP Multipath:

- By default, BGP selects a single path as the best path and installs it in the IP routing table.
- If configured, a BGP router can select up to six identical BGP routes as the best routes.

```
router(config-router)# maximum-paths number
```

- With **maximum-paths** configured, a BGP router can select several identical EBGp routes as the best routes and install them in the IP routing table for load-sharing purposes.

A router can use either only [IBGP](#) routes or only [EBGP](#) routes for load balancing but not both at the same time because this could lead to routing loops. BGP multipath can be enabled separately for IBGP and EBGp routes. A router can install up to six BGP routes in the IP routing table. The actual type of load sharing (per-session or per-packet) that occurs between the routes depends on the switching mode that is used.

To control the maximum number of parallel routes that an IP routing protocol can support, use the **maximum-paths** command in address family or router configuration mode. To restore the default value, use the **no** form of this command.

```
maximum-paths number
```

Syntax Description

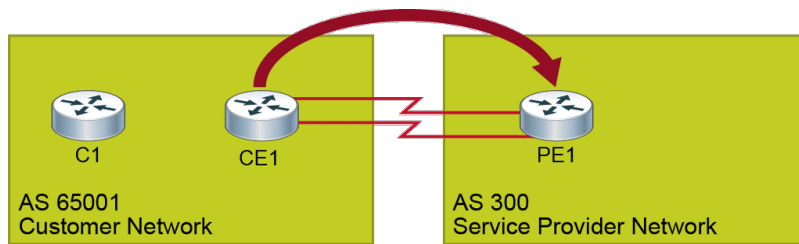
Parameter	Description
<i>number</i>	Maximum number of parallel routes that an IP routing protocol installs in a routing table, in the range of 1 to 6.



Load sharing between alternative BGP routes is achieved only if the EBGp routes are identical according to all BGP route selection rules and **maximum-paths** is configured with a value larger than 1.

Load Sharing with EBGP Multihop

When two adjacent routers have multiple links between them, you can configure the [EBGP](#) session from loopback interface to loopback interface. In this case, you must use the **ebgp-multihop** option to make the [BGP](#) session go into the active state. There must be static or dynamic routing in use to provide both routers with information on how to reach the loopback interfaces of each other. Otherwise, their EBG session does not complete establishment.



EBGP Multihop:

- Used when parallel links between routers exist. Loopback interfaces have to be used for BGP peering.
- Because of recursive lookup, load sharing toward a BGP destination always occurs if there are several equal-cost paths to the BGP next hop.
- By default, EBG neighbors must be directly connected.
- The **ebgp-multihop** [*TTL*] command declares an EBG neighbor to be distant (several hops away).

Routing to the loopback interface of the neighboring router is required to establish the EBG session. It is also used in the recursive lookup when the router installs the routes in its routing table. The two routes to the loopback interface of the neighboring router should be equivalent for load sharing to occur.

After configuration, one single EBG session is established between the two routers. This session is used to exchange the routing information. There is only one BGP route to each destination, and it has a next hop that refers to the loopback interface of the other router.

The load balancing is automatic as a result of the recursive route lookup of BGP next hops. The IP routing table will contain a single routing entry to reach the customer prefix but the Cisco Express Forwarding (CEF) table (which contains the results of the recursive lookup) will have two equal-cost paths to reach the customer prefix.

By default, EBG neighbors must be directly connected. A router verifies that an EBG neighbor is reachable as directly connected over one of the router interfaces before the session goes into the active state. For an EBG session, IP packets that carry the [TCP](#) segments with BGP information are also sent using a [TTL](#) value set to the value 1. This value means that they cannot be routed.

The **ebgp-multihop** neighbor configuration command changes this behavior. Although the neighbor is several hops away, the session goes into the Active state, and packets start to be exchanged. The TTL value of the IP packets is set to a value larger than 1. If no value is specified on the command line, 255 is used.

Use the **ebgp-multihop** command when you are establishing EBG sessions between loopback interfaces for load-sharing purposes. You must take great care when using **ebgp-multihop**. Proper packet forwarding relies on all the intermediate routers along the path to the EBG peer to make the correct forwarding decision. If the intermediate routers have a correct path to the EBG peer but a wrong path to the final destination, the packet may get into a routing loop.

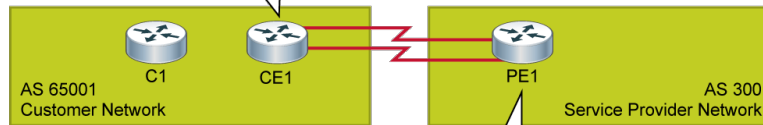
Load Sharing with EBG Multihop Example

In the figure, the customer network and the [ISP](#) network are connected via single router on each side. Two parallel links are used.

```

CE1(config)# interface Loopback 0
CE1(config-if)# ip address 10.0.0.1 255.255.255.0
CE1(config-if)# interface Serial 0
CE1(config-if)# ip address 172.16.33.3 255.255.255.0
CE1(config-if)# interface Serial 1
CE1(config-if)# ip address 172.16.34.3 255.255.255.0
CE1(config-if)# exit
CE1(config)# router bgp 65001
CE1(config-router)# neighbor 10.0.3.1 remote-as 300
CE1(config-router)# neighbor 10.0.3.1 update source Loopback 0
CE1(config-router)# neighbor 10.0.3.1 ebgp-multihop 2
CE1(config-router)# exit
CE1(config)# ip route 10.0.3.1 255.255.255.255 172.16.33.33
CE1(config)# ip route 10.0.3.1 255.255.255.255 172.16.34.34

```



```

PE1(config)# interface Loopback 0
PE1(config-if)# ip address 10.0.3.1 255.255.255.0
PE1(config-if)# interface Serial 0
PE1(config-if)# ip address 172.16.33.33 255.255.255.0
PE1(config-if)# interface Serial 1
PE1(config-if)# ip address 172.16.34.34 255.255.255.0
PE1(config-if)# exit
PE1(config)# router bgp 300
PE1(config-router)# neighbor 10.0.0.1 remote-as 65001
PE1(config-router)# neighbor 10.0.0.1 update source Loopback 0
PE1(config-router)# neighbor 10.0.0.1 ebgp-multihop 2
PE1(config-router)# exit
PE1(config)# ip route 10.0.0.1 255.255.255.255 172.16.33.3
PE1(config)# ip route 10.0.0.1 255.255.255.255 172.16.34.3

```

In this case, only one EBGP session is configured between the customer and provider routers. The session should be established from the loopback interface on one router to the loopback interface on the other.

Each of the two edge routers has two static host routes that point to the loopback interface on the other router. The EBGP session is established from loopback to loopback using **ebgp-multihop**.

The customer receives an EBGP route from the ISP with the next hop set to 10.0.3.1. The **CE** router performs a recursive lookup and finds that it can reach 10.0.3.1 via 172.16.33.33 and via 172.16.34.34. These two routes are equivalent. Therefore, the route to the final destination is installed in the routing table of the customer router using both paths.

Depending on the switching mode in use, load sharing is done per packet, per destination, or per source and destination pair.

In this example, link-level procedures ensure that if one of the links goes down, the corresponding static link goes down. All BGP routes in the routing table that rely on the static route to the link that went down are invalidated. However, the BGP routes in the routing table that rely on the remaining link are still valid and used.

Summary

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

- When a customer has multiple connections to a single ISP and the link-level procedures cannot detect a link failure, a routing protocol is required. For security reasons, this routing protocol must be BGP.
- Customer does not necessarily use a public AS number, it can also use private AS numbers that range from 64512 to 65535. BGP must remove private AS numbers.
- The ISP should advertise a default route to the customer through BGP. The provider should also use incoming filters to ensure that the customer advertises only the correct address space and AS number.
- The BGP Support for Dual AS Configuration for Network AS Migrations feature allows you to merge a secondary AS under a primary AS without disrupting customer peering sessions.
- By announcing portions of its address space, a customer can use maximum paths and EBGP multihop to provide load sharing over multiple links.

Connecting a Multihomed Customer to Multiple Service Providers

Overview

When a customer requires the maximum redundancy in its network design, it should implement a multihomed strategy that uses multiple service providers. This configuration requires specific considerations to be implemented properly. Addressing and [AS](#) number selection are important considerations that affect the network implementation. It is also important to understand how to configure routing protocols so that customer backup or load-sharing requirements are met.

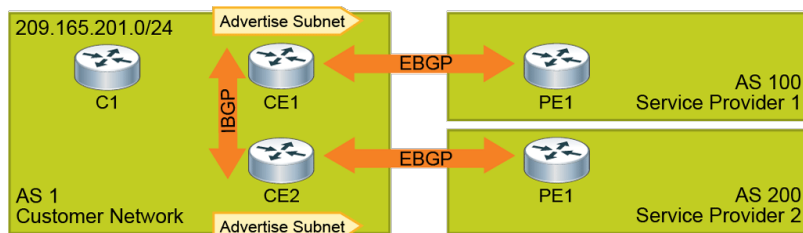
In this lesson, you will discuss multiple connections between a customer and multiple service providers for backup and load-sharing purposes. Including the [BGP](#) characteristics that are used to configure such connectivity. You will also discuss address selection, private AS number translation, and configuration of the network to support either backup links or load sharing.

Upon completing this lesson, you will be able to:

- Describe BGP configuration characteristics that are used to establish routing between a multihomed customer and multiple service providers
- Describe addressing strategies that are available to a multihomed customer that is connected to multiple service providers
- Describe AS numbering strategies that are available to a multihomed customer that is connected to multiple service providers
- Describe the operation of AS number translation
- Describe how you can implement a typical backup setup between a multihomed customer and multiple service providers in a BGP environment
- Describe the use of BGP attributes to influence inbound link selection in customer networks that are multihomed to multiple service providers
- Describe how you can implement load sharing between a multihomed customer and multiple service providers in a BGP environment

BGP Configuration for Multihomed Customers

The highest level of redundancy to network failures is achieved in network designs that connect the customer network to two different service provider networks. Customers use this option when the requirement for resilient Internet connectivity is very high. This requirement also involves duplication of equipment to make the customer network fully redundant.



- BGP is run between the customer and the service provider.
- The customer advertises the allocated address space into BGP. It is recommended to use a public AS number.
- The service provider has to deploy inbound BGP filters.
- The service provider advertises the following to the customer:
 - Default route, default route + local networks, or full Internet routing table

Because the customer has to use [PI address](#) space, it is the responsibility of the customer to advertise the customer address space. The only routing option that can be used in this scenario is [BGP](#) because service providers run only BGP with customers. It is not enough to detect link failures or a failure in the remote router by link-level procedures. Failures that occur beyond the directly connected router must also be detected, and the only means of detecting these failures is by using a routing protocol. The only routing protocol that is suited for the Internet environment is BGP. Correctly configured, BGP takes care of rerouting in the following situations:

- Link failure between the customer network and the network of one of the Internet service providers.
- Edge router failure on either the customer or the service provider side.
- Link failure or router failure within the customer network that causes the [CE](#) router to lose connectivity with the customer network core. This situation requires correct configuration of route advertisement as described in an earlier lesson.
- Link failure or router failure within the [ISP](#) network that causes the [PE](#) router to lose connectivity with the rest of the Internet

Multihomed customers have multiple permanent links to different ISPs. The links should terminate in different edge routers in the customer network. Otherwise, one of the major advantages, resilience to router failure, is lost.

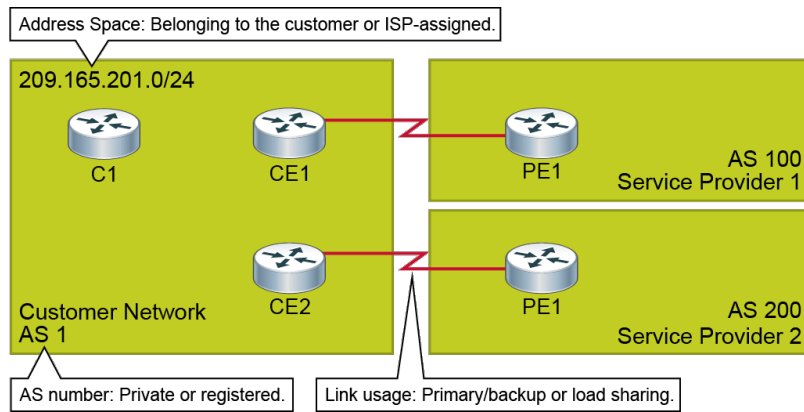
Multihomed customers should use BGP with both ISPs. The customer should advertise its address space to both providers. The route advertisement should be configured in both CE routers. The advertisement should be conditioned at the edge routers by the appropriate route policies leading toward the core of the customer network. This setup is analogous to the one configured when you are connecting a multihomed customer network to a single service provider.

The customer should take care not to move any routing information between the two ISPs. It must use outgoing filters to prevent any route that is received from one of the ISPs from being propagated to the other. Otherwise, the customer network appears as a transit network between the two ISPs.

Both ISPs must apply filters on the incoming BGP information from the customer. These filters will protect themselves and the rest of the Internet from errors in the BGP configuration of the customer. Each of the service providers must accept routes from the customer that indicate networks within the customer address space only. [AS](#) path filter lists should be implemented on the PE routers to allow incoming routes only if they have the correct AS path attribute value. If the incoming filters on the ISP edge router accept customer routes, then the service provider should propagate those routes to the rest of the Internet.

Both ISPs must provide the customer with at least some BGP routes. Depending on the customer requirements, the volume of BGP routes the ISP provides can vary. It

can be the default route only, the default route with local routes as well, or the full Internet routing table.



Before configuring the multihomed network, you need to consider the following questions:

- Should any of the links be used as primary and the others as backup?
- Should both links share the load?
- What address space is the customer using? Is the customer address space [PA](#) or [PI](#)?
- What AS number is the customer using? (Is the customer using a public or a private AS number?)

Multihomed Customer Address Space Selection

If the customer has its own address space, it should announce it to both service providers. Both providers are responsible for propagating the customer routes to the rest of the Internet without doing any summarization.

- Provider-independent address space:
 - If the customer owns the address space, there should be no limitations regarding announcing it to both service providers.
- Provider-assigned address space:
 - If the customer uses ISP-assigned small address blocks, then there is no purpose in using BGP to provide redundant connectivity. NAT is easier to implement and solves the problem of the reverse path.

However, if the customer uses a small block of addresses that one of the [ISPs](#) assigned, an alternative design that does not involve [BGP](#) exist. The customer can use two different [PA address](#) spaces and do [NAT](#). With NAT, the router translates traffic going out over one of its connections to one of the PA addresses. If the traffic goes out the other way, the addresses are translated to an address from the address space of the other provider.

Multihomed Customer AS Number Selection

The use of [BGP](#) requires an [AS](#) number. The preferred option is to use a registered, public AS number. However, registered AS numbers are assigned only to those who really need them because public AS numbers are a scarce resource. A customer with BGP sessions to multiple [ISPs](#) must use a registered, public AS number. A customer that is connected to only one ISP does not require a public AS number. In that case, a private AS number in the range from 64512 to 65535 is sufficient.

Registered, public AS number (recommended):

- Preferred option, but difficult to get.
- Does not require ISPs to assign a private AS number.
- Consistent routing information in the Internet.

Private AS number (discouraged):

- Easier to get (even easier with AS translation).
 - One private AS number: The customer has to be able to use the same private AS number with multiple providers.
 - Multiple private AS numbers: Each service provider assigns a private AS number to the customer and the customer then uses one of them internally; the others have to be translated.
- Causes inconsistent routing information.

Whenever the customer is assigned a public AS number, there are no conflicts in the BGP setup because the number is guaranteed to be unique within the Internet. Both, the customer and the service provider make route announcements without tampering with the AS path. As a result, the service provider propagates a consistent AS path information to the rest of the Internet.

In cases where the customer does not have a public AS number, it must use a private AS number. Because private AS numbers are not propagated to the Internet, several network administrators can make this assignment independently of each other. In this case, AS numbers are reused, which conserves AS number space. A service provider normally assigns private AS numbers to its customers. This arrangement ensures that unique private AS numbers are used among the customers of a single ISP.

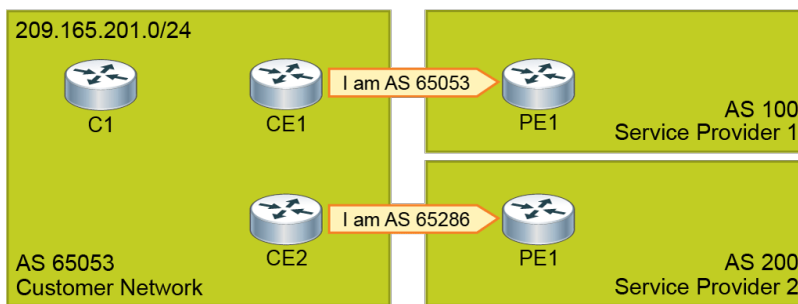
When a customer is going to be multihomed and the private AS number that has already been assigned by one of the ISPs comes in conflict with an AS number that has been assigned by the other ISP, the customer needs to consider renumbering the customer AS. If the two service providers can reach a common agreement on which private AS number the multihomed customer should use, renumbering is a solution. If no common agreement can be made or if renumbering, for some reason, is not an option, AS translation must be configured on the customer network.

No router should ever propagate private AS numbers to the rest of the Internet. An ISP can keep track of the private AS numbers that it has assigned to its customers and avoid reuse or conflicts within that scope. However, as soon as the scope is widened to include other ISPs, conflicts will happen. Each ISP, therefore, removes private AS numbers from the AS path before sending routes outside its own AS.

When the routes with the private AS numbers removed are propagated to the rest of the Internet, the AS path looks as though the routes were originated within the public AS of the ISP. All information about the private AS lying behind the public AS is lost. In the case of a multihomed customer, the customer routes are first propagated into each of the autonomous systems of its ISPs. In the next step, the private AS number is removed from the routes, as the routes are propagated to the rest of the Internet. Now the customer routes appear to be originating in the ASs of both ISPs. To an outside observer, there is now an AS path inconsistency because it looks as though the same route belongs to different ASs.

AS Number Translation

The figure shows a case where a customer is multihomed but forced to use two private AS numbers (for example, because of the scarcity of public AS numbers).



- On one EBGP adjacency, the real AS number is used.
- On the other EBGP adjacency, the AS number is translated to the one assigned by the second ISP.

In the figure, Service Provider 1 has assigned the private AS number 65053 to the customer. Service provider 2 did not agree to use this private AS number when connecting to the customer. Instead, service provider 2 has assigned the private AS number 65286.

The customer now has two different private AS numbers: 65053 and 65286. The customer decides to use 65053 internally. All router [BGP](#) configuration lines have 65053 as the AS number. The customer uses AS number 65286 only when establishing the [EBGP](#) session to AS 200.

In the example, Service Provider 1 (AS 100) has an EBGP session to the customer where the AS number 65053 is used at the customer end. Service provider 2 (AS 200) has an EBGP session to the customer where the AS number 65286 is used at the customer end. Translation between these two private AS numbers takes place in the [CE](#) router as part of the EBGP session to AS 200.

```
router(config-router)# neighbor ip-address local-as private-as [no-  
prepend [replace-as]]
```

- Optionally, the service providers can assign two different private AS numbers to the customer.
- Internally, the customer can use an ISP-assigned AS number or even any other private AS number.
- Externally, the customer appears as one private AS to ISP 1 and as a different private AS to ISP 2.
- **Note:** When you are using this option, the AS path of the customer network contains two AS numbers. The ISP has to adapt the incoming AS path filters.

The **neighbor ip-address local-as private-as** router configuration command is used to indicate the AS number that the local router uses as its local AS number in the BGP Open message. The remote router is assumed to have an EBGP session to the indicated local AS.

Internally, the customer network uses another private AS number. When routes are sent to the neighbor, the internal AS number is automatically prepended in the AS path first, and then the specified local AS number is prepended as well. As a consequence, the [ISP](#) receives the routes with an AS path with both AS numbers in it. The ISP has to adapt its incoming filter-lists as a result of this situation.

When a keyword **no-prepend** is used, router does not prepend the local autonomous system number to any routes received from the EBGP neighbor.

When a keyword **replace-as** is used, router replaces the real autonomous system number with the local autonomous system number in the eBGP updates. The autonomous system number from the local BGP routing process is not prepended.



Some service providers might be unwilling to change their AS path input filters, leaving the customer no other option than to use a public AS number or to connect to a single ISP with a private AS number.

Primary and Backup Link Selection

When using [BGP](#) on multiple links between a customer and several service provider networks, the customer is solely responsible for controlling the use of the links between them for outgoing traffic. The customer chooses whether to use these links in a primary and backup, or a load-sharing configuration.

In the primary and backup scenario, only the default route is required from each service provider. The customer implements routing policies.

- **Outgoing link selection:**
 - You can use the same solution as with multihomed customers connected to one service provider.
 - A higher local preference for the default route comes from the primary service provider.
- **Incoming link selection:**
 - You cannot use the MED because it can be sent only to the neighboring AS and no farther.
 - You must use other means such as BGP communities or AS path prepending to achieve incoming link selection.
- Only the default route is required from both service providers.

If one link is primary and the other is backup, the customer can use the local preference attribute in the configuration to direct all outgoing traffic over the primary link. The customer selects a primary service provider for outgoing traffic by setting a higher local preference for routes that are received from the preferred [ISP](#). This configuration is no different than the configuration that is used for customers with multiple connections running BGP to a single service provider.

Controlling the load distribution of incoming traffic over multiple links is more difficult in the multihomed scenario when links to multiple service providers are used. You cannot use the [MED](#) when the customer connects to multiple providers because the updates are sent to two different [ASs](#). Recall that the MED is used only when you compare routes that are received from a single directly connected AS over two parallel links.

Therefore, route selection decisions will most likely use the AS path attribute and prefer the route with the shortest AS path length. The customer can configure AS path prepending of routes that are sent to the backup service provider to lengthen the AS path. The returning traffic originating from the Internet will therefore choose a shorter path through the primary service provider.

BGP Incoming Link Selection

To remove incoming traffic from the backup link, the customer must influence route selection in the backup AS. The backup ISP must be forced to prefer the primary path to reach the customer network, although this choice means selecting a route with a longer AS path.

BGP communities:

- Customer sets the appropriate BGP community attribute on updates that are sent to the backup ISP.
- Requires the ISP to translate the BGP community attribute to a local preference attribute that is lower than the default value of 100.
- May not work in all situations.

AS path prepending:

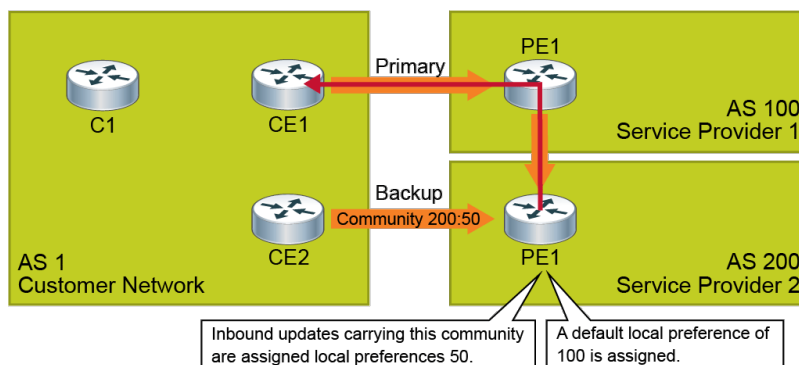
- Multiple copies of customer AS number prepended to the AS path to lengthen the AS path sent over the backup link.
- Customer not dependent on the provider configuration.
- Always works.

One way to influence route selection is to use local preference in the network of the backup ISP. Using local preference creates an administrative scalability issue if each customer requires its use, because the ISP must maintain the configuration.

One scalable way of setting local preference in an ISP network is to use communities. The customer sets a well-known community value on the routes that are sent to the backup ISP. The ISP recognizes the community and sets the local preference for these routes. This solution is available only if the ISP has implemented and announced the use of communities. If communities and a local preference setting are used, route selection occurs only if there are alternative routes to compare.

Another way of influencing route selection in the backup ISP is to do AS path prepending before sending the advertisement to the backup ISP. When the customer sends routes over the backup link, multiple copies of its own AS number are prepended to the AS path of each route. The backup ISP receives the routes and makes normal route-selection decisions. No special weight or local preference settings are used; the BGP route selection is based exclusively on the AS path length. No special configuration is required in the service provider network.

BGP Incoming Link Selection Using BGP Communities



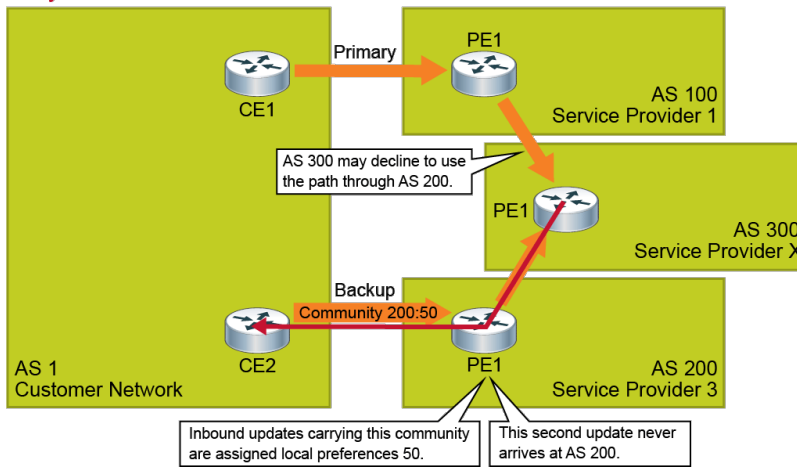
In the example here, the backup service provider 2 (AS 200) has defined the meaning of community 200:50. When AS 200 receives routes with this community, the local preference is set to 50.

The customer AS 1 is advertising the route over the primary link without any communities. It is received by AS 100 and propagated to AS 200. When AS 200 receives the route via AS 100, there is no community set. AS 200 therefore assigns the default local preference value of 100.

The customer is also advertising the route over the backup link. However, in this case, the route has the community 200:50 set. When AS 200 receives this route, it recognizes the community, and the local preference value is set to 50.

Route selection is now performed in AS 200. The route that has been received via AS 100 is preferred based on the local preference values.

May Not work?



Even when the use of communities is correctly configured, the desired load distribution may not always be achieved. As this example shows, AS 200 does not always receive the primary route, although nothing is wrong in the network.

The customer AS 1 sends routes with community 200:50 over the backup link to AS 200. AS 200 receives the routes and sets the local preference to 50. If AS 200 over some time selects the directly connected path to AS 1 as the best, it propagates the route to AS 300. As the route is propagated over the EBGP session between AS 200 and AS 300, the local preference value that is used within AS 200 is lost.

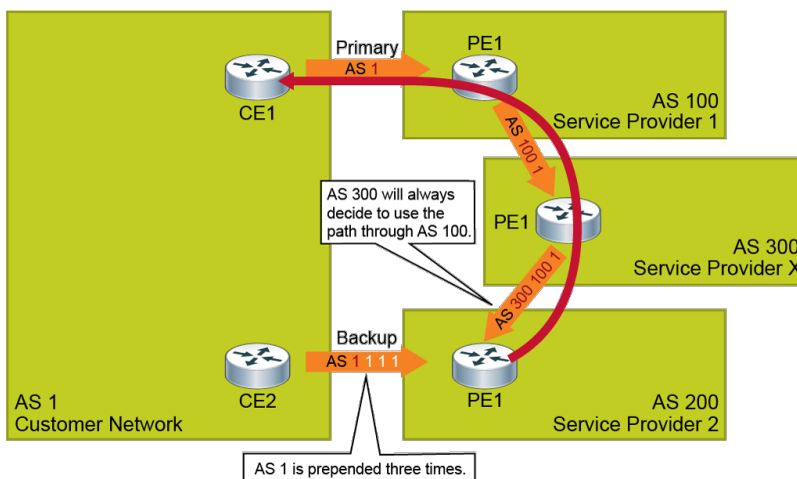
AS 300 does not have any use for the community 200:50 because this community is defined and implemented only within AS 200. Potentially, the community value can also be stripped off during BGP route propagation.

Customer AS 1 also sends the routes over the primary link to AS 100. The routes are propagated to AS 300, which now sees two alternative routes to the destination networks within AS 1. The router in AS 300 does not use weight nor local preference as criteria for reaching AS 1. Both alternatives have equal AS path lengths.

The route selection decision that will be made in AS 300 is hard to predict, but the outcome definitely influences the route selection decision that was made in AS 200. If AS 300 prefers the route to the customer network via AS 200 for any reason, then the second-best alternative via AS 100 and the primary link is never propagated to AS 200.

In this case, AS 200 never sees the primary path and has to stick to the backup link and announce the route to AS 300. The network has reached a steady state when the traffic uses the backup link although the primary link is available.

BGP Incoming Link Selection Using AS Path Prepending



In this example, the customer AS 1 is performing AS path prepending on the backup link. Three copies of the customer AS number (1) are prepended to the AS path. As the route goes out over the EBGP session, BGP prepends the local AS number to the AS path attribute. AS 200 receives routes from AS 1 over the backup link with an

AS path length of four. This is the original AS 1 plus three prepended copies that the [CE](#) router applied to the AS path attribute.

The customer advertises networks without AS path prepending over the primary link. AS 100 receives routes with an AS path length of one and propagates these routes to AS 300, which then receives them with an AS path length of two.

If, for a short time, AS 300 received the customer routes via AS 200, the AS path length of those routes would have been five. In that case, AS 300 selects the route from AS 100 as the best and propagates it to AS 200.

AS 200 now sees both alternatives. The customer routes that have been received directly from the customer have an AS path length of four. The routes that have been received via AS 300 have an AS path length of three. Because no weight or local preference is configured in this example, AS 200 selects the route via AS 300 as the best.

The desired result, to have all traffic enter the customer network via the primary link, is now achieved.

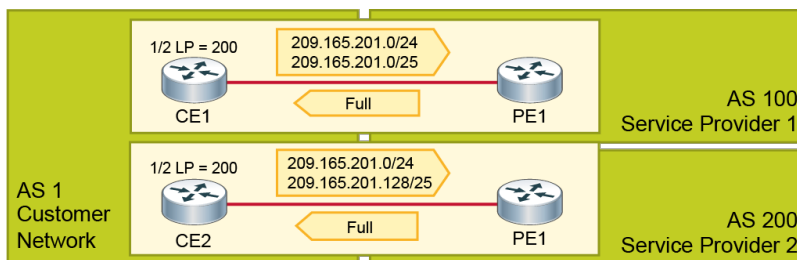


If the backup ISP is implementing incoming AS path filters for this customer with the length of the AS path equal to one, the ISP has to change the configuration of the AS path filter for the customer. The ISP can either create a new filter, allowing multiple copies of the customer AS number only for this customer, or use regular expression variables to create a common filter for all customers that belong to one peer group.

Load Sharing with Multiple Providers

Load sharing over links to two different ISPs can be compared to doing load sharing over two parallel links to a single ISP. The only difference is that there is only one option that is available to control incoming traffic. Controlling load distribution of the outgoing traffic is configured in the same way as when a multihomed customer connects to a single service provider.

- **Load sharing for *outgoing* traffic:**
 - You can use the same solution as with multihomed customers that are connected to one service provider.
 - A higher local preference for half of the routes comes from one service provider. A higher local preference for the other half of the routes comes from the other service provider.
- **Load sharing for *incoming* traffic:**
 - The only load-sharing option that you can use in this setup is to separate the address space into two or more smaller address blocks.
 - Some traffic analysis is needed to fine-tune the address space separation according to link bandwidths.
 - You should use AS path prepending to ensure symmetric routing as well as backup for noncontiguous address blocks.



- The customer divides the address space into two announcements.

The customer can control the load distribution of incoming traffic based on the traffic destination. The customer divides its address space into several announcements. One announcement is sent to one of the ISPs. Another announcement is sent to the other ISP. For backup purposes, the customer announces the entire address space to both ISPs. The ISPs now use the most-explicit route rule. As long as both links are up, traffic with destinations within one part of the customer address space is routed over one of the links. Traffic to the other part is routed over the other link.

It is very difficult to predict the volume of traffic that will be directed to each part of the customer address space. You should monitor the results of changing route updates by watching the load on the links before and after implementing the change. If the load distribution is not satisfactory, you can further modify the division of the address space. You must then check the load on the links again and further fine-tune the configuration.

A customer may decide to use both the division of address space into several advertisements and AS path prepending together. Some part of the customer address space may be advertised by the customer network with a longer AS path over one of the links to fine-tune the load. Also, there may be cases where there are noncontiguous subnets that cannot be divided because the prefixes would be too long. These subnets are evenly distributed between the links in a primary/backup configuration.

Summary

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

- A customer that is multihomed to multiple BGP service providers must advertise its address space to both ISPs, but has to make sure that it does not become a transit network.
- Customers who are connected to multiple ISPs must use an AS number that all ISPs agree to.
 - You can use AS number translation to prepend a different AS number to the AS path, which allows the customer to use a single private AS number in the network.
- Primary and backup connectivity:
 - Outgoing route selection is achieved by using local preference.
 - Incoming route selection is achieved by using either BGP communities or AS path prepending.
- The key for load-sharing is to divide the customer address space.

Module Summary

Overview

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

- Customer requirements influence the planning of connectivity between a customer and an ISP, including:
 - Physical connection type (the redundancy provided)
 - IP addressing
 - AS numbering
- Static or dynamic routing can be used between a customer and a service provider.
 - **Static routing:** Customers with a single or multiple connections to the service provider (link detection and device failure can be detected)
 - **Dynamic routing** (BGP): Multihomed customers.
- When using dual links and BGP, BGP attribute manipulation is required to achieve the required traffic distribution pattern:
 - Primary and backup link
 - Load sharing

References

For additional information, refer to these resources:

- Cisco Systems, Inc. *Sample Configuration for BGP with Two Different Service Providers (Multihoming)*.
http://www.cisco.com/en/US/tech/tk365/technologies_configuration_example09186a008009456d.shtml
- Cisco Systems, Inc. *Removing Private Autonomous System Numbers in BGP*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13756-32.html>
- Cisco Systems, Inc. *How BGP Routers Use the Multi-Exit Discriminator for Best Path Selection*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13759-37.html>
- Cisco Systems, Inc. *Configuring the BGP Local-AS Feature*.
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13761-39.html>
- Cisco Systems, Inc. *Load Sharing with BGP in Single and Multihomed Environments: Sample Configurations*.
http://www.cisco.com/en/US/tech/tk365/technologies_configuration_example09186a00800945bf.shtml

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. If a customer requires extra bandwidth and redundancy, which approach is preferred? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ A single permanent connection to one ISP.

☐ Permanent connections to more than one ISP.

☐ Dial-up connections to more than one ISP.

☐ Multiple permanent connections to one ISP.

1. Which two routing options could be used when connecting a customer to a service provider? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ static routing

☐ OSPF

☐ IS-IS

☐ BGP

1. A customer would like to connect to a service provider. Which two requirements should be considered before deciding on a type of connectivity? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ application availability

☐ redundancy

☐ cost ownership

☐ flexibility

1. Which type of redundancy do multiple permanent connections that provide load-sharing configuration display? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ link

☐ equipment

☐ service provider

☐ routing protocol

1. In a customer-to-provider routing scheme, which method of routing is preferred because of its lower complexity? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

•
policy-based routing

•
dynamic routing

•
content routing

•
static routing

1. Why is it that with multiple permanent connections to more than one ISP, the use of dynamic routing with BGP is required? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

- When one of the connections is lost, the link level detects this loss and places the interface in a down state.

- Monitoring of the link status cannot detect a problem inside one of the ISP networks.

- Static routes detect problems inside one of the ISP networks.

- It is not required, and static routing may be used.

1. What can be done when a customer is assigned only a very small subnet of public addresses? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

- Purchase more addresses as required.

- Use NAT.

- Add a service provider.

- Add links to the same service provider.

1. What are two different addressing schemes that customers use to connect to a service provider? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ provider-independent

☐ customer-independent

☐ provider-assigned

☐ customer-assigned

1. Which two of the following criteria are required for a customer to be multihomed to multiple ISPs? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

- ☐ The customer must have a public AS number.
- ☐ The customer must have a private AS number.
- ☐ The customer must run BGP with both of its ISPs.
- ☐ The customer must run BGP with one ISP and may use static routing with the other.

1. Floating static routes do not demand any additional configuration to work with BGP. True or false? (Source: "Implementing Customer Connectivity Using Static Routing")

☒ True

☐ False

1. When static routing is implemented between the customer network and the ISP network, the customer must announce a default route. True or false? (Source: "Implementing Customer Connectivity Using Static Routing")

☐ True

☐ False

1. What are two requirements for being able to use static routing as part of installing redundant connections between the customer network and a single service provider network? (Choose two.)
(Source: "Implementing Customer Connectivity Using Static Routing")

- ☐ The router must be able to detect a link failure.
- ☐ The default route must be announced using the customer IGP.
- ☐ If one link goes down, the interface must remain in an up state.
- ☐ The customer IGP must continue to advertise the static default route.

1. A customer route that should not be announced to the rest of the Internet is marked using what? (Source: "Implementing Customer Connectivity Using Static Routing")

☐ a route tag

☐ the export community

☐ the no-export community

☐ the public address filter

1. When you are designing a static route propagation in a service provider network, which three steps must you take? (Choose three.)
(Source: "Implementing Customer Connectivity Using Static Routing")

- ☐ Assign a tag to each combination of services.
- ☐ Configure a community that matches defined tags.
- ☐ Redistribute static routes into BGP through a route map.
- ☐ Identify all possible combinations of services that are offered to a customer.

1. What does a route map assign that will be used by other routers within a network? (Source: "Implementing Customer Connectivity Using Static Routing")

a tag

community values

public addressing

QoS

1. Which three key pieces of information can you derive from the following router command output? (Choose three.) (Source: "Implementing Customer Connectivity Using Static Routing")CE2#
show ip bgp 209.165.201.0
BGP routing table entry for 209.165.201.0/24, version 7
Paths: (2 available, best #1, not advertised to EBGp peer)
Advertised to non peer-group peers:
10.3.0.5
Local
0.0.0.0 from 0.0.0.0 (10.3.0.6)
Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best
Community: 1:31000 no-export
Local
10.3.0.2 (metric 128) from 10.3.0.5 (1.0.0.2)
Origin incomplete, metric 0, localpref 100, valid, internal
Originator: 1.0.0.2, Cluster list: 10.3.0.5
Community: 1:31000 no-export

▪
The primary link has come back up, so the backup router now sees two alternate routes.

▪
The primary link has not come back up, but the backup router still sees two alternate routes.

▪
The first route is the route that the router itself has redistributed into BGP using the floating static route. This route is locally sourced by the AS and has been assigned a weight value of 32768.

▪
The second route is the one that has been received by IBGP from the primary edge router. The AS also sources this route, but no weight value is assigned.

1. Which two things can you do to overcome the problems that occur when a floating static route is redistributed into BGP? (Choose two.)
(Source: "Implementing Customer Connectivity Using Static Routing")

- ☐ You must raise the weight value.
- ☐ You must lower the weight value.
- ☐ You must set the AD at a higher value than all other routes.
- ☐ You must assign local preference values, giving the floating static route a lower local preference value than the primary route.

1. What are the three characteristics of using static routes during load sharing of outgoing traffic? (Choose three.) (Source: "Implementing Customer Connectivity Using Static Routing")

- ☐ Outgoing traffic load sharing is easy to achieve.
- ☐ Each customer router uses the closest customer edge router as the exit point.
- ☐ Balanced load sharing is achieved if the customer edge routers are collocated.
- ☐ Local preference values must be assigned, giving the floating static route a lower local preference value than the primary route.

1. What are the three responsibilities of the customer when the customer has multiple connections to a single service provider? (Choose three.) (Source: "Connecting a Customer to a Single Service Provider")

- ☐ Customer edge routers must run IBGP between them.
- ☐ The customer must advertise a default route.
- ☐ The customer must conditionally advertise its assigned address space into BGP.
- ☐ The customer edge routers must run EBGp with the provider.

1. Given the following router command output, which method has been used to influence return traffic in a primary/backup link implementation for the customer with multiple connections to a service provider? (Source: "Connecting a Customer to a Single Service Provider")ISP# **show ip bgp**

BGP table version is 5, local router ID is 10.0.33.34

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
* 10.10.20.0/24	192.168.63.3	1000		0 100	100 i
*>	192.168.64.4	2000		0 400	100 i
*> 30.30.30.0/24	192.168.63.3	0		0 100	I
*> 40.40.40.0/24	192.168.64.4	0		0 400	I

MED

local preference

weight

AS path prepending

1. What are three responsibilities of the provider router when supporting a customer with multiple connections? (Choose three.)
(Source: "Connecting a Customer to a Single Service Provider")

☐ The provider must advertise a default route to the customer through BGP.

☐ The provider must filter customer routes to verify that proper addressing is used.

☐ The provider must remove the private AS number, if it is in use by the customer.

☐ The provider must configure new AS path filters to allow AS path prepending; otherwise, a primary/backup link cannot be established.

1. What will occur if private AS numbers are advertised to the Internet?
(Source: "Connecting a Customer to a Single Service Provider")

- The Internet will not be able to route packets.

- Internet routers could drop routes based on BGP loop-prevention mechanisms.

- Customer load balancing will not function.

- Customer configurations for the primary/backup link using AS path prepending will not function.

1. Which two BGP configurations are required to properly implement a backup solution for a customer that is connected to a single provider via multiple connections? (Choose two.) (Source: "Connecting a Customer to a Single Service Provider")

☐ The customer should set local preference to influence outgoing route selection.

☐ The customer should set the weight attribute to influence outgoing path selection.

☐ The customer should set the MED on each route to influence return path selection.

☐ The customer should configure AS path prepending to ensure proper outgoing path selection.

1. A customer router has been configured with maximum paths set to a value of 4. Given the following router command output, over how many links will the router need to perform load balancing? (Source: "Connecting a Customer to a Single Service Provider")CE1# **show ip bgp**

BGP table version is 5, local router ID is 10.0.33.34

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

Network	Next Hop	Metric	LocPrf	Weight	Path
* 10.10.20.0/24	192.168.63.3			0 300 100	100 i
*>	192.168.64.4			0 400 100	i
*	192.168.65.5			0 500 100	i
*> 30.30.30.0/24	192.168.63.3	0		0 300	i
*> 40.40.40.0/24	192.168.64.4	0		0 400	i

The router will use only the path marked as "best" by BGP.

The router will perform load balancing over two paths to reach network 10.10.20.0/24.

The router will perform load balancing over three paths to reach network 10.10.20.0/24.

There is not enough information to determine the correct answer.

1. Which three methods can you use to provide load sharing over network links between a customer and a single provider? (Choose three.) (Source: "Connecting a Customer to a Single Service Provider")

- ☐ Advertising of split addressing space to the provider.
- ☐ Configuring **ebgp-multihop** between the customer and the provider.
- ☐ Using the BGP **maximum-paths** command to perform load balancing over parallel links.
- ☐ Configuring multiple static routes that point to the provider.

1. Why is it not required to configure maximum paths under the BGP routing process when load balancing is being performed because the **ebgp-multihop** command has been configured? (Source: "Connecting a Customer to a Single Service Provider")

- By default, BGP will perform load balancing over up to four paths, configurable up to six.

- The static route or IGP process is responsible for load balancing in this configuration.

- Configuring multihop enables maximum paths equal to the TTL setting of the **neighbor ebgp-multihop** command.

- Configuring **ebgp-multipath** is a required component of **ebgp-multihop** load balancing.

1. Which two of the following characteristics accurately describe the BGP Support for Dual AS Configuration for Network AS Migrations feature? (Choose two.) (Source: "Connecting a Customer to a Single Service Provider")

- ☐ Allows you to merge a secondary AS under a primary AS without disrupting customer peering sessions.
- ☐ Allows a router to appear, to external peers, as a member of primary AS during the AS migration.
- ☐ Allows a router to appear, to external peers, as a member of secondary AS during the AS migration.
- ☐ Eliminates the possibility that routing loops can be created.

1. A multihomed customer is using AS number 65550 internally. The customer is connected to two different providers. ISP1 (in AS 200) has assigned the customer an AS number of AS 65101. ISP2 (in AS 300) has assigned the customer an AS number of AS 65201. Given that the customer will use AS number translation for its internal AS, what is the AS path attribute (attached to routes that originated in the customer network) that will be displayed on a router in the network of ISP2? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

65550 i

65201 i

65201 65550 i

300 65201 i

1. Which three methods can you use to provide load sharing over network links between a multihomed customer and multiple providers? (Choose three.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

- ☐ Advertising of split addressing space to the provider.
- ☐ Configuring of multiple static routes that point to the provider.
- ☐ Using the BGP **maximum-paths** command to perform load sharing over parallel links.
- ☐ AS path prepending to fine-tune the load-sharing configuration.

1. What are three BGP configuration characteristics of a multihomed customer that is connected to multiple providers? (Choose three.)
(Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ The customer announces assigned addressing to its providers through BGP.

☐ The customer announces a default route to its network through BGP.

☐ The provider announces a default route, local routes, or full Internet routing to the customer via BGP.

☐ The customer configures outbound filters to prevent its network from becoming a transit area.

1. A multihomed customer is using AS number 1024 and is connected to two different providers (ISP1: AS 200 and ISP2: AS 300). The customer has configured the MED to ensure a proper return path so that ISP1 is the primary provider and ISP2 is the backup provider. Unfortunately, return traffic continues to use the backup link. What is a possible cause of this problem? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

• The backup provider is ignoring the MED attribute on received routes.

• The MED attribute cannot be sent to the backup provider because it is local to AS 1024 only.

• The customer has not set the proper BGP communities to allow the primary and backup providers to correctly set the MED attribute.

• The MED cannot be used in this scenario, because it will not be advertised to providers upstream of ISP2.

1. What are three important considerations for customers that wish to connect to multiple providers? (Choose three.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ The customer has to consider whether to use PA or PI address space.

☐ The customer has to decide whether to use static routes or BGP to connect to upstream providers.

☐ The customer has to decide whether to use a public AS number or a private AS number scheme.

☐ The customer has to decide whether to perform load sharing or use a primary/backup implementation over redundant links.

1. Which AS number selection is the best possible choice for a customer that is multihomed to multiple providers? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ A single public AS number.

☐ A single private AS number.

☐ Two private AS numbers that are used in conjunction with AS number translation.

☐ Multiple private AS numbers, one used internally by the customer and the others used in conjunction with AS number translation for each provider.

1. Given the following router command output, which two methods have been configured to influence return traffic in a primary/backup link for this multihomed customer? (Choose two.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")
- ```
ISP# show ip bgp
BGP table version is 5, local router ID is 10.0.33.34
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
* 10.10.20.0/24 192.168.63.3 2000 0 100 100 i
*> 192.168.64.4 0 300 100 i
*> 30.30.30.0/24 192.168.63.3 0 0 300 100 i
*> 40.40.40.0/24 192.168.64.4 0 0 100 i
```

☐ MED

☐ local preference

☐ split address advertisement

☐ AS path prepending

1. You are a customer connected to two service providers with one link to each provider. You would like to achieve primary and backup traffic distribution. Which two BGP attributes do you have to manipulate? (Choose two.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ local preference for incoming traffic

☐ local preference for outgoing traffic

☐ AS path for incoming traffic

☐ MED for incoming traffic



1. A multihomed customer can get only one private AS number, regardless the number of service providers. True or false? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ True

☐ False

## Answer Key

1. If a customer requires extra bandwidth and redundancy, which approach is preferred? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ A single permanent connection to one ISP.

☐ Permanent connections to more than one ISP.

☐ Dial-up connections to more than one ISP.

☐ Multiple permanent connections to one ISP.

1. Which two routing options could be used when connecting a customer to a service provider? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☒ static routing

☐ OSPF

☐ IS-IS

☒ BGP

1. A customer would like to connect to a service provider. Which two requirements should be considered before deciding on a type of connectivity? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ application availability

☒ redundancy

☐ cost ownership

☒ flexibility

1. Which type of redundancy do multiple permanent connections that provide load-sharing configuration display? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ link

☐ equipment

☐ service provider

☐ routing protocol

1. In a customer-to-provider routing scheme, which method of routing is preferred because of its lower complexity? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ policy-based routing

☐ dynamic routing

☐ content routing

☒ static routing

1. Why is it that with multiple permanent connections to more than one ISP, the use of dynamic routing with BGP is required? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

When one of the connections is lost, the link level detects this loss and places the interface in a down state.

Monitoring of the link status cannot detect a problem inside one of the ISP networks.

Static routes detect problems inside one of the ISP networks.

It is not required, and static routing may be used.

1. What can be done when a customer is assigned only a very small subnet of public addresses? (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☐ Purchase more addresses as required.

☐ Use NAT.

☐ Add a service provider.

☐ Add links to the same service provider.



1. What are two different addressing schemes that customers use to connect to a service provider? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

☒ provider-independent

☐ customer-independent

☒ provider-assigned

☐ customer-assigned

1. Which two of the following criteria are required for a customer to be multihomed to multiple ISPs? (Choose two.) (Source: "Understanding Customer-to-Provider Connectivity Requirements")

- ☒ The customer must have a public AS number.
- ☐ The customer must have a private AS number.
- ☒ The customer must run BGP with both of its ISPs.
- ☐ The customer must run BGP with one ISP and may use static routing with the other.

1. Floating static routes do not demand any additional configuration to work with BGP. True or false? (Source: "Implementing Customer Connectivity Using Static Routing")

☒ True

☐ False

1. When static routing is implemented between the customer network and the ISP network, the customer must announce a default route. True or false? (Source: "Implementing Customer Connectivity Using Static Routing")

☐ True

☐ False

1. What are two requirements for being able to use static routing as part of installing redundant connections between the customer network and a single service provider network? (Choose two.)  
(Source: "Implementing Customer Connectivity Using Static Routing")



The router must be able to detect a link failure.



The default route must be announced using the customer IGP.



If one link goes down, the interface must remain in an up state.



The customer IGP must continue to advertise the static default route.

1. A customer route that should not be announced to the rest of the Internet is marked using what? (Source: "Implementing Customer Connectivity Using Static Routing")

☐ a route tag

☐ the export community

☐ the no-export community

☐ the public address filter

1. When you are designing a static route propagation in a service provider network, which three steps must you take? (Choose three.)  
(Source: "Implementing Customer Connectivity Using Static Routing")

- ☒ Assign a tag to each combination of services.
- ☐ Configure a community that matches defined tags.
- ☒ Redistribute static routes into BGP through a route map.
- ☒ Identify all possible combinations of services that are offered to a customer.

1. What does a route map assign that will be used by other routers within a network? (Source: "Implementing Customer Connectivity Using Static Routing")

a tag

community values

public addressing

QoS



1. Which three key pieces of information can you derive from the following router command output? (Choose three.) (Source: "Implementing Customer Connectivity Using Static Routing")CE2#  
**show ip bgp 209.165.201.0**  
BGP routing table entry for 209.165.201.0/24, version 7  
Paths: (2 available, best #1, not advertised to EBGp peer)  
Advertised to non peer-group peers:  
10.3.0.5  
Local  
0.0.0.0 from 0.0.0.0 (10.3.0.6)  
Origin incomplete, metric 0, localpref 100, weight 32768, valid, sourced, best  
Community: 1:31000 no-export  
Local  
10.3.0.2 (metric 128) from 10.3.0.5 (1.0.0.2)  
Origin incomplete, metric 0, localpref 100, valid, internal  
Originator: 1.0.0.2, Cluster list: 10.3.0.5  
Community: 1:31000 no-export

- ☒ The primary link has come back up, so the backup router now sees two alternate routes.
- ☐ The primary link has not come back up, but the backup router still sees two alternate routes.
- ☒ The first route is the route that the router itself has redistributed into BGP using the floating static route. This route is locally sourced by the AS and has been assigned a weight value of 32768.
- ☒ The second route is the one that has been received by IBGP from the primary edge router. The AS also sources this route, but no weight value is assigned.

1. Which two things can you do to overcome the problems that occur when a floating static route is redistributed into BGP? (Choose two.)  
(Source: "Implementing Customer Connectivity Using Static Routing")

- ☐ You must raise the weight value.
- ☒ You must lower the weight value.
- ☐ You must set the AD at a higher value than all other routes.
- ☒ You must assign local preference values, giving the floating static route a lower local preference value than the primary route.

1. What are the three characteristics of using static routes during load sharing of outgoing traffic? (Choose three.) (Source: "Implementing Customer Connectivity Using Static Routing")

- ☒ Outgoing traffic load sharing is easy to achieve.
- ☒ Each customer router uses the closest customer edge router as the exit point.
- ☒ Balanced load sharing is achieved if the customer edge routers are collocated.
- ☐ Local preference values must be assigned, giving the floating static route a lower local preference value than the primary route.

1. What are the three responsibilities of the customer when the customer has multiple connections to a single service provider? (Choose three.) (Source: "Connecting a Customer to a Single Service Provider")

- ☒ Customer edge routers must run IBGP between them.
- ☐ The customer must advertise a default route.
- ☒ The customer must conditionally advertise its assigned address space into BGP.
- ☒ The customer edge routers must run EBGp with the provider.

1. Given the following router command output, which method has been used to influence return traffic in a primary/backup link implementation for the customer with multiple connections to a service provider? (Source: "Connecting a Customer to a Single Service Provider")ISP# **show ip bgp**

BGP table version is 5, local router ID is 10.0.33.34

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

| Network          | Next Hop     | Metric | LocPrf | Weight | Path  |
|------------------|--------------|--------|--------|--------|-------|
| * 10.10.20.0/24  | 192.168.63.3 | 1000   |        | 0 100  | 100 i |
| *>               | 192.168.64.4 | 2000   |        | 0 400  | 100 i |
| *> 30.30.30.0/24 | 192.168.63.3 | 0      |        | 0 100  | I     |
| *> 40.40.40.0/24 | 192.168.64.4 | 0      |        | 0 400  | I     |

MED

local preference

weight

AS path prepending

1. What are three responsibilities of the provider router when supporting a customer with multiple connections? (Choose three.)  
(Source: "Connecting a Customer to a Single Service Provider")

☒ The provider must advertise a default route to the customer through BGP.

☒ The provider must filter customer routes to verify that proper addressing is used.

☒ The provider must remove the private AS number, if it is in use by the customer.

☐ The provider must configure new AS path filters to allow AS path prepending; otherwise, a primary/backup link cannot be established.

1. What will occur if private AS numbers are advertised to the Internet?  
(Source: "Connecting a Customer to a Single Service Provider")

The Internet will not be able to route packets.

Internet routers could drop routes based on BGP loop-prevention mechanisms.

Customer load balancing will not function.

Customer configurations for the primary/backup link using AS path prepending will not function.

1. Which two BGP configurations are required to properly implement a backup solution for a customer that is connected to a single provider via multiple connections? (Choose two.) (Source: "Connecting a Customer to a Single Service Provider")

☒ The customer should set local preference to influence outgoing route selection.

☐ The customer should set the weight attribute to influence outgoing path selection.

☒ The customer should set the MED on each route to influence return path selection.

☐ The customer should configure AS path prepending to ensure proper outgoing path selection.



1. A customer router has been configured with maximum paths set to a value of 4. Given the following router command output, over how many links will the router need to perform load balancing? (Source: "Connecting a Customer to a Single Service Provider")CE1# **show ip bgp**

BGP table version is 5, local router ID is 10.0.33.34

Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal

Origin codes: i - IGP, e - EGP, ? - incomplete

| Network          | Next Hop     | Metric | LocPrf | Weight    | Path  |
|------------------|--------------|--------|--------|-----------|-------|
| * 10.10.20.0/24  | 192.168.63.3 |        |        | 0 300 100 | 100 i |
| *>               | 192.168.64.4 |        |        | 0 400 100 | i     |
| *                | 192.168.65.5 |        |        | 0 500 100 | i     |
| *> 30.30.30.0/24 | 192.168.63.3 | 0      |        | 0 300     | i     |
| *> 40.40.40.0/24 | 192.168.64.4 | 0      |        | 0 400     | i     |

The router will use only the path marked as "best" by BGP.

The router will perform load balancing over two paths to reach network 10.10.20.0/24.

The router will perform load balancing over three paths to reach network 10.10.20.0/24.

There is not enough information to determine the correct answer.

1. Which three methods can you use to provide load sharing over network links between a customer and a single provider? (Choose three.) (Source: "Connecting a Customer to a Single Service Provider")

- ☒ Advertising of split addressing space to the provider.
- ☒ Configuring **ebgp-multihop** between the customer and the provider.
- ☒ Using the BGP **maximum-paths** command to perform load balancing over parallel links.
- ☐ Configuring multiple static routes that point to the provider.

1. Why is it not required to configure maximum paths under the BGP routing process when load balancing is being performed because the **ebgp-multihop** command has been configured? (Source: "Connecting a Customer to a Single Service Provider")

- By default, BGP will perform load balancing over up to four paths, configurable up to six.

- The static route or IGP process is responsible for load balancing in this configuration.

- Configuring multihop enables maximum paths equal to the TTL setting of the **neighbor ebgp-multihop** command.

- Configuring **ebgp-multipath** is a required component of **ebgp-multihop** load balancing.

1. Which two of the following characteristics accurately describe the BGP Support for Dual AS Configuration for Network AS Migrations feature? (Choose two.) (Source: "Connecting a Customer to a Single Service Provider")

☒ Allows you to merge a secondary AS under a primary AS without disrupting customer peering sessions.

☐ Allows a router to appear, to external peers, as a member of primary AS during the AS migration.

☒ Allows a router to appear, to external peers, as a member of secondary AS during the AS migration.

☐ Eliminates the possibility that routing loops can be created.

1. A multihomed customer is using AS number 65550 internally. The customer is connected to two different providers. ISP1 (in AS 200) has assigned the customer an AS number of AS 65101. ISP2 (in AS 300) has assigned the customer an AS number of AS 65201. Given that the customer will use AS number translation for its internal AS, what is the AS path attribute (attached to routes that originated in the customer network) that will be displayed on a router in the network of ISP2? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

65550 i

65201 i

65201 65550 i

300 65201 i

1. Which three methods can you use to provide load sharing over network links between a multihomed customer and multiple providers? (Choose three.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

- ☒ Advertising of split addressing space to the provider.
- ☐ Configuring of multiple static routes that point to the provider.
- ☒ Using the BGP **maximum-paths** command to perform load sharing over parallel links.
- ☒ AS path prepending to fine-tune the load-sharing configuration.

1. What are three BGP configuration characteristics of a multihomed customer that is connected to multiple providers? (Choose three.)  
(Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☒ The customer announces assigned addressing to its providers through BGP.

☐ The customer announces a default route to its network through BGP.

☒ The provider announces a default route, local routes, or full Internet routing to the customer via BGP.

☒ The customer configures outbound filters to prevent its network from becoming a transit area.

1. A multihomed customer is using AS number 1024 and is connected to two different providers (ISP1: AS 200 and ISP2: AS 300). The customer has configured the MED to ensure a proper return path so that ISP1 is the primary provider and ISP2 is the backup provider. Unfortunately, return traffic continues to use the backup link. What is a possible cause of this problem? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

The backup provider is ignoring the MED attribute on received routes.

The MED attribute cannot be sent to the backup provider because it is local to AS 1024 only.

The customer has not set the proper BGP communities to allow the primary and backup providers to correctly set the MED attribute.

The MED cannot be used in this scenario, because it will not be advertised to providers upstream of ISP2.



1. What are three important considerations for customers that wish to connect to multiple providers? (Choose three.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☒ The customer has to consider whether to use PA or PI address space.

☐ The customer has to decide whether to use static routes or BGP to connect to upstream providers.

☒ The customer has to decide whether to use a public AS number or a private AS number scheme.

☒ The customer has to decide whether to perform load sharing or use a primary/backup implementation over redundant links.

1. Which AS number selection is the best possible choice for a customer that is multihomed to multiple providers? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ A single public AS number.

☐ A single private AS number.

☐ Two private AS numbers that are used in conjunction with AS number translation.

☐ Multiple private AS numbers, one used internally by the customer and the others used in conjunction with AS number translation for each provider.

1. Given the following router command output, which two methods have been configured to influence return traffic in a primary/backup link for this multihomed customer? (Choose two.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")ISP# **show ip bgp**  
BGP table version is 5, local router ID is 10.0.33.34  
Status codes: s suppressed, d damped, h history, \* valid, > best, i - internal  
Origin codes: i - IGP, e - EGP, ? - incomplete
- | Network          | Next Hop     | Metric | LocPrf | Weight | Path      |
|------------------|--------------|--------|--------|--------|-----------|
| * 10.10.20.0/24  | 192.168.63.3 | 2000   |        | 0      | 100 100 i |
| *>               | 192.168.64.4 |        |        | 0 300  | 100 i     |
| *> 30.30.30.0/24 | 192.168.63.3 | 0      |        | 0 300  | 100 i     |
| *> 40.40.40.0/24 | 192.168.64.4 | 0      |        | 0      | 100 i     |

☐ MED

☐ local preference

☒ split address advertisement

☒ AS path prepending

1. You are a customer connected to two service providers with one link to each provider. You would like to achieve primary and backup traffic distribution. Which two BGP attributes do you have to manipulate? (Choose two.) (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☐ local preference for incoming traffic

☒ local preference for outgoing traffic

☒ AS path for incoming traffic

☐ MED for incoming traffic

1. A multihomed customer can get only one private AS number, regardless the number of service providers. True or false? (Source: "Connecting a Multihomed Customer to Multiple Service Providers")

☒ True

☐ False

# Scaling Service Provider Networks

## Introduction

In standard [BGP](#) implementations, all BGP routers within an [AS](#) must be fully meshed so that all external routing information can be distributed among the other routers that reside within the AS. Therefore, within an AS, all routers must establish [TCP](#) sessions with all other BGP routers. As the AS grows, scalability challenges arise because of an ever-increasing number of TCP sessions and demands for router [CPU](#) and memory resources.

This module discusses network scalability concerns that are common to large, complex service provider networks. The module also discusses BGP route reflectors as scalability mechanism that allows you to steer away from BGP full-mesh requirements and improve network scalability by reducing the number of TCP sessions that are required within an AS. Also discussed in this module are the Cisco IOS commands that are needed to configure and monitor BGP route reflectors.

Upon completing this module, you will be able to:

- Describe common routing scalability issues in service provider networks
- Describe the function of route reflectors in a BGP environment
- Configure proper operation of route reflectors to modify IBGP split-horizon rules in an existing IBGP network

# Scaling IGP and BGP in Service Provider Networks

## Overview

Properly scaling [IP addressing](#), [IGPs](#), and [BGP](#) is a common area of concern to all service providers and can be the difference between a successful and a problematic BGP implementation. Service provider networks are complex and must meet the administrative policy and routing demands of the internal network, different customers, and other providers. Proper scaling is crucial to the success of the network. Interactions between IGPs and the BGP, specifically when you support internal routing, customer connectivity, and transit traffic (and the administrative policies that match), can be quite complex. Furthermore, the large number of prefixes that are required to support full Internet routing requires administrators to fully characterize IGP and BGP interactions for internal networks and customers alike.

This lesson discusses network scalability concerns common to large, complex service provider networks. Included in this lesson is a description of a typical [ISP](#) network and discussion of the propagation of internal and customer routing information. Also scaling considerations for IGPs and BGP, and scaling of IP addressing in service provider networks are described.

Upon completing this lesson, you will be able to:

- Describe the basic structure of service provider networks
- Describe the propagation of internal and customer routes in service provider networks
- Describe proper scaling of IGPs and BGP in service provider networks
- Describe scaling issues that are relevant to IP addressing in ISP networks

## Common Service Provider Network

The common service provider network runs [EBGP](#) or static routing with customers. EBGP is always used as the routing protocol between different service providers.

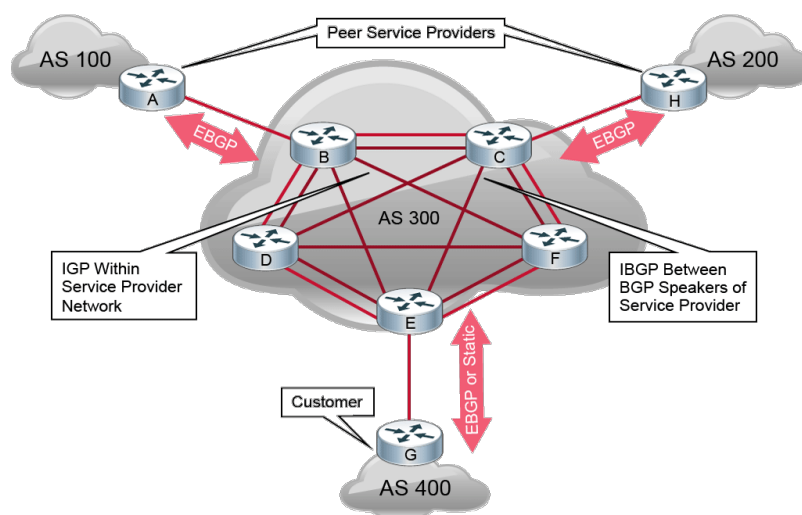
- Runs BGP or static routing with customer.
- Exchanges routes with other service providers via BGP.
- Runs IBGP between its own BGP speakers.
- Runs one instance of IGP (OSPF or IS-IS).
  - IGP used for internal routes only.

[IBGP](#) is required in the provider network because all EBGP-speaking routers in an [AS](#) must exchange external routes via IBGP. Also, non-EBGP speakers are required to take part in the IBGP exchange if they are in a transit path and forward packets based on destination IP addresses.

The service provider network also runs an [IGP](#). The protocols of choice are [OSPF](#) and [IS-IS](#). The IGP is used for two purposes:

- Provides IP connectivity between all IBGP speakers so that [TCP](#) sessions for IBGP can be established between [BGP](#)-speaking routers
- Provides optimal routing to the BGP next-hop address

A single IGP should be used within the entire AS. This setup facilitates effective packet forwarding from the ingress router to egress routers. The IGP is configured to carry internal routes only, including internal links and loopback addresses of the routers. For performance and scalability reasons, no customer routes or external routes should be injected into the IGP.



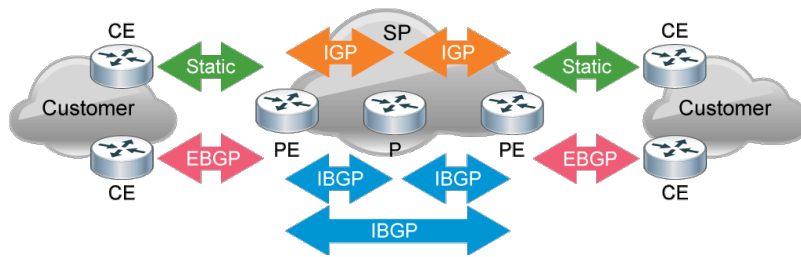
The typical service provider network consists of a network core that connects various edge devices. Some of the edge devices connect customers; others connect to other service providers.

The edge devices that connect to other service providers use EBGP to exchange routing information. The edge devices that connect customers use either static routing or EBGP.

Unless [MPLS](#) is configured on the service provider backbone, routers in a transit path are also required to have full routing information. Therefore, these routers take part in the IBGP routing exchange.

An IGP is also required within the service provider network. The IGP is used to carry internal routes, including the loopback interface addresses of IBGP-speaking routers. The IGP provides reachability information to establish IBGP sessions and to perform the recursive routing lookup for the BGP next hop.





- PE routers use EBGP or static routing with CE routers.
- PE and P routers use full-mesh IBGP routing.
- The provider core IGP is a single instance of IS-IS or OSPF and is used only within the service provider core network.
- Optimal routing between PEs is desired.

CE routers connect via IP to PE routers. In many cases, the PE routers use static routing to customer networks. The PE routers advertise static routes to the rest of the service provider network and to other autonomous systems using BGP. PE and P routers use full-mesh IBGP routing to exchange BGP routing information.

Service providers use BGP routing with the customer when redundancy requires the use of a routing protocol.

The service provider backbone typically uses a single instance of either IS-IS or OSPF as its IGP. The IGP is used within the provider backbone only. The provider backbone exchanges no IGP routing information with customer routers or with routers in other autonomous systems.

## Route Propagation in Service Provider Networks

It is important to avoid sending any unnecessary routing information in the [IGP](#). The IGP performs best if it carries as few routes as possible. Optimally, the IGP should contain only information about [BGP](#) next hops and routes that are internal to the service provider network, enabling the establishment of [IBGP](#) sessions.

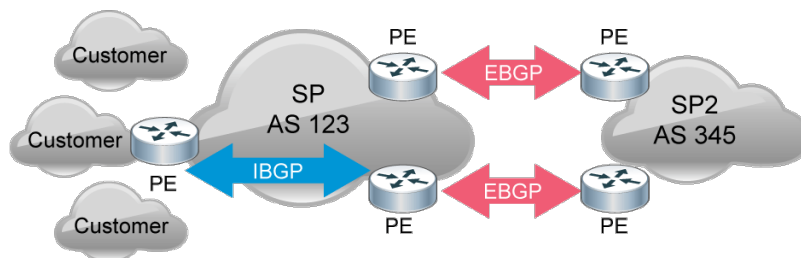
- BGP route propagation
  - BGP carries customer routes.
  - BGP carries other provider routes.
- IGP route propagation
  - IGP is responsible only for the resolution of BGP next hop and internal routes
- Do not redistribute BGP into IGP.
  - IGP performance and convergence time suffer if many are carried.
  - No IGP is capable of carrying full Internet routes.
  - A full Internet routing table has exceeded 600,000 routes.

All other routing information should be carried in BGP, which is designed to scale for large volumes of routing information. Customer routes and the routes from other service providers should be carried in BGP. These routes should not be propagated from BGP into the provider IGP.

IGP performance and convergence time suffer if the IGP carries a larger number of routes. The design goal should be to minimize the volume of routing information that the IGP carries. Naturally, the number of route flaps is also reduced as the number of routes is reduced.

BGP scales to a much larger volume of routing information because of the inherent qualities of the design of BGP. Potentially, the BGP routers of the service provider can receive the full Internet routing table, which has exceeded 600,000 routes. You should therefore never redistribute the routing information that the BGP received into the IGP, because no IGP is capable of carrying several tens of thousands of routes.

## Routing Information Exchange with Other Service Providers



BGP is used to exchange routing information with upstream service providers:

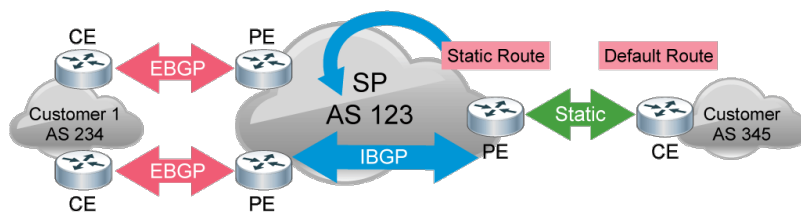
- Service provider sends summary of service provider-owned address space to upstream service provider.
- Service provider sends prefixes owned by customers using independent address space.
- Upstream service provider sends full Internet routing table to the service provider.

The figure shows how routes are exchanged between service providers.

Provider edge routers use BGP to exchange routing information with other service provider networks for redundancy and scalability reasons.

Routing information from customers is received at [PE](#) routers using [EBGP](#) and then propagated using [IBGP](#) to the rest of the service [P-network](#). At another PE router, the routing information is further propagated to a different service provider using [EBGP](#) with other autonomous systems. The service provider sends a summary of service provider-owned space to upstream service providers, along with prefixes owned by the customer using provider-independent address space. The upstream service provider sends the full Internet routing table to the service provider, which allows the service provider to implement a primary/backup scenario or a load-balancing scenario when peering with multiple upstream service providers.

## Routing Information Exchange with Customers

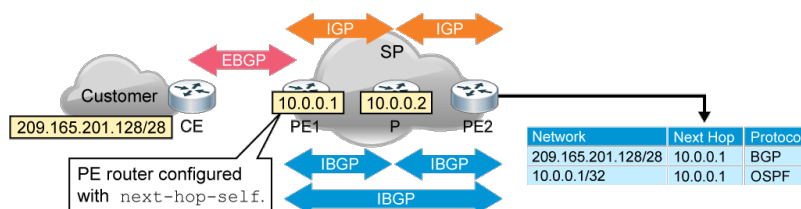


- BGP with customer:
  - Customer advertises its address space.
  - Service provider advertises default route, service provider-owned routes and default route, or full Internet routing table.
- Static routing with customer:
  - Customer uses default route.
  - Service provider uses static route on the **PE** router for customer address space. Static route is redistributed into BGP on the PE router.

The figure shows how customers exchange routes with the service provider. In this case, the customer advertises its address space to the service provider. The service provider advertises to the customer either the default route, service provider-owned routes and the default route, or the full Internet routing table, based on the customer requirements.

When the customer does not require redundancy, static routing is used. In this case, the customer configures a static default route that points to the PE router of the service provider. The service provider configures a static route for the customer network that points to the **CE** router. The PE router redistributes static routes into BGP. The service P-network then uses BGP to propagate the information to the rest of the service P-network using IBGP. The service provider also advertises customer routing information to other autonomous systems using BGP.

## Next-Hop Resolution



- The core IGP of the service provider should carry information only about backbone links and loopback addresses.
- **next-hop-self** feature on the PE routers removes the need to include access links in IGP, and thus prevents route flapping if access link flaps.

The figure shows why IGP is needed in the service provider core network. The IGP used in the service provider core should carry information only about backbone links and loopback addresses. The service provider should use BGP to carry all other information.

Use the BGP **next-hop-self** command when BGP routing is exchanged with the customer or other service providers. Using the **next-hop-self** command results in the BGP next hop being set to the loopback address of the service provider edge router and not to the access link address of the customer. The IGP can then be relieved of the burden of carrying information about the access link. The benefit of not carrying customer link information is that a flapping access link will not disturb the service provider IGP.

As shown in the routing table on the PE2 router, the next hop for the customer network points to the address of the PE1 route. This is usually the **IP address** of the loopback interface. Because the PE1 and PE2 routers are not directly connected, an IGP routing protocol is needed that carries information about core links and loopbacks across the service provider network.

## Scaling Service Provider Routing Protocols

[BGP](#) and [IGP](#) have different responsibilities in service provider networks.

IGP responsibilities:

- Carrying route to BGP next hop.
- Providing optimal path to next hop.
- Converging to alternate path so that BGP peering is maintained.

BGP responsibilities:

- Generating BGP update.
- Scaling BGP policies.
- Scaling IBGP mesh.
- Reducing impact of flapping routes.

The IGP is responsible for the following:

- Carrying routes to the BGP next hops to facilitate recursive routing.
- Providing an optimal path to the next hop, thereby optimizing packet flow toward all BGP destinations.
- Converging to an alternate path in the case of lost links or routers in a redundant network (which should be quick so that BGP sessions are not lost).

The BGP is responsible for the following:

- Generating BGP updates about reachable and unreachable networks.
- Implementing and scaling the BGP routing policy, which can be quite cumbersome in large service provider networks with many [EBGP](#)-speaking routers.
- Implementing and scaling [IBGP](#) sessions between all BGP-speaking routers in the [AS](#).
- Reducing the impact of individual flapping routes through route summarization.

### Scaling IGP

In scaling an IGP, it is important to limit the number of routes that the IGP carries. Optimally, the IGP carries only loopback interfaces and internal links.

- Loopbacks and internal links carried only.
- Good addressing structure within the POP required.
- Loopback addresses taken out of a different address space and not summarized.
- Summarization of internal link addresses on POP level.
- Optimal routes to loopbacks needed only (with proper summarization).

The number of routes that the IGP carried can be even further reduced with route summarization. However, care must be taken because loopback addresses should never be summarized. Route summarization always introduces the risk of suboptimal routing and should be carefully planned, because it is important that recursive routing lookup always uses optimal routing to the next hop. Also, in an [MPLS](#) environment, an [LSP](#) must be unbroken between edge routers, and summarizing loopback interfaces will break the LSP.

Internal links can always be summarized because they are not used as BGP next-hop addresses. To facilitate proper route summarization, internal links and loopback interfaces on a router should be assigned addresses from two different address spaces. Also, the internal links of a router should be assigned addresses depending upon which [POP](#) the routers belong to.

If implemented correctly, all internal router links in one POP can be summarized at the POP level and injected into the core as a single route. But, all router loopback addresses within the POP are still propagated into the core as individual host routes, giving optimal routing to all loopback interfaces.

### Scaling BGP

Scaling BGP includes scaling both IBGP and EBGP as well as scaling BGP updates and routing table size.

- BGP policy scaling:

- The AS routing policy should be unitary and easy to maintain.
- This goal is achieved by reusing the same configuration in all EBGp-speaking routers.
- IBGP mesh scaling:
  - Avoid unnecessary duplicate updates over a physical link.
  - Full-mesh IBGP is not needed since there are other technologies and features available
- Updates and table size scaling:
  - Route summarization is the key to scalability.

The task of scaling BGP actually involves three different and independent scaling tasks:

- **BGP policy scaling:** The AS routing policy should be unitary and easy to maintain. Different edge routers of the same AS should not use different policies and thereby advertise different routes to neighboring autonomous systems. Regardless of which router is currently active, the same routing policy should be in place. Administratively, replication of the same routing policies requires the same configuration lines in several edge routers.
- **IBGP mesh scaling:** All BGP-speaking routers must be updated with consistent IBGP information. In the traditional BGP approach, ensuring consistent routing information was achieved by establishing a full mesh of IBGP sessions between all routers within the AS. An IBGP full mesh is certainly not scalable, and several tools are now available to achieve the same results without the full mesh.
- **Updates and table size scaling:** The number of routes in the routing table and the number of updates that are sent and received represent the third scaling task. Route summarization is the key to this scalability.

## Scaling Service Provider Addressing

Internal [IP addressing](#) in the service provider core network can be simplified to reduce the use of public addresses and to simplify configuration.

This list shows how to scale addressing in Service Provider Core Network:

- Private or public IP addresses can be used.
- Private addresses on core links and loopbacks display private IP addresses in a traceroute when run from customers.
  - MPLS with TTL propagation disabled solves the traceroute issue.
- Private addresses on loopbacks and core links call for careful external routing to prevent advertisement of private addresses to customers or upstream service providers.
- Otherwise, use public addresses in service provider core networks.

For [IPv4](#), public or private IP addresses can be used on core links. However, using private addresses has some drawbacks. Private addresses on the provider internal links will cause trouble for the traceroute application. When the **traceroute** command is executed from the customer, the customer will see private IP addresses in the traceroute printout. Using [MPLS](#) without [TTL](#) propagation in the service [P-network](#) can easily overcome the traceroute problem with private addresses. If these functions are used, the provider network will appear as a single hop to the traceroute application. The intermediate routers will be invisible and thus can use private addresses.

Using private addresses on the service provider router loopback interfaces is possible. However, you must take care not to advertise any private addresses to any other [AS](#).

A rule of safety is to prevent the announcement of any private addresses by using prefix lists that are applied on outgoing updates to external neighbors. The same prefix list mechanism can also be used on the provider edge routers. It can be used to prevent accepting private addresses from any other AS if the other AS, by mistake, announces private addresses.

## Summary

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

- Service providers most commonly use integrated IS-IS and OSPF as interior gateway protocols and BGP as the exterior gateway protocol.
- BGP is used to carry customer routes while IGPs are used to carry service provider internal prefix reachability information.
- BGP allows ISP clients to acquire information about all or some networks reachable through the ISP.
- ISP can use static routing or BGP to direct traffic going to the customers to the correct links.
- Next-hop-self can be used to avoid redistributing transit segments into IGP on iBGP neighbors.
- When BGP networks grow, various actions must be taken to make them scalable, for iBGP scalability use route reflectors.
- When IP networks grow, several aspects of addressing need to be considered to reduce sizes of routing tables and to avoid consuming too many addresses.

# Introducing and Designing Route Reflectors

## Overview

Large [BGP](#) networks cannot properly scale without relying on performance-enhancing tools such as route reflectors and confederations. Route reflectors enable BGP routing information to be distributed in a fashion that does not require a physical fully meshed network. Network overhead is reduced by decreasing the number of [TCP](#) connections that are required to distribute routing information and by lessening router [CPU](#) and memory requirements.

This lesson introduces BGP route reflectors by explaining why they improve BGP scalability. Modified split-horizon rules, applied when you are using route reflectors, are also discussed. The lesson also describes the various redundancy mechanisms that are used with route reflectors, including route reflector clusters. The lesson introduces network design rules that you should follow when implementing a network with BGP route reflectors. It also lists the potential issues that can arise if the network design rules are not adhered to. The lesson concludes by describing the concept of hierarchical route reflectors.

Upon completing this lesson, you will be able to:

- Explain the need for BGP route reflectors in BGP transit backbones
- Explain how route reflectors modify traditional IBGP split-horizon rules
- Explain the benefits of deploying redundant route reflectors
- Explain how route reflector clusters prevent loops in the deployment of route reflectors in redundant configurations
- Describe additional route reflector mechanisms that have been designed to prevent routing loops
- List the network design rules for implementing BGP route reflectors
- List the potential issues that can arise if you do not follow the route reflector network design rules
- Explain the function of hierarchical route reflectors



## IBGP Scalability Issues in a Transit AS

Classic [IBGP](#) split-horizon rules specify that updates that are received on an [EBGP](#) session should be forwarded on all IBGP and EBGP sessions. However, updates that are received on an IBGP session should be forwarded only to all EBGP sessions. This rule requires a [BGP](#) boundary router to be able to send routing updates to all other BGP-speaking routers in its own [AS](#) directly through a separate IBGP session to each of them.

IBGP requires a full mesh between all BGP-speaking routers.

- Large number of TCP sessions.
- Unnecessary duplicate routing traffic.

Solutions:

- Route reflectors modify IBGP split-horizon rules.
- BGP confederations modify IBGP AS-path processing.

The primary reason for the IBGP split-horizon rule is to avoid routing information loops within the AS. If the information that is received through an IBGP session is forwarded on other IBGP sessions, the information might come back to the originator. This information is then forwarded again in a never-ending loop. The originator would not detect the loop because no BGP attributes are changed on IBGP sessions.

The general design rule in classic IBGP is to have a full mesh of IBGP sessions. But a full mesh of IBGP sessions between  $n$  number of routers would require  $(n * (n - 1)) / 2$  IBGP sessions. For example, a router with an AS that contains 10 routers would require  $(10 * (10 - 1)) / 2 = 45$  IBGP sessions. Imagine the number of sessions (and the associated router configuration) that would be required for a single AS containing 500 routers.

Every IBGP session uses a single [TCP](#) session to another IBGP peer. An update that must be sent to all IBGP peers must be sent on each of the individual TCP sessions. If a router is attached to the rest of the network over just a single link, this single link has to carry all [TCP/IP](#) packets for all IBGP sessions. This requirement results in multiplication of the update over the single link.

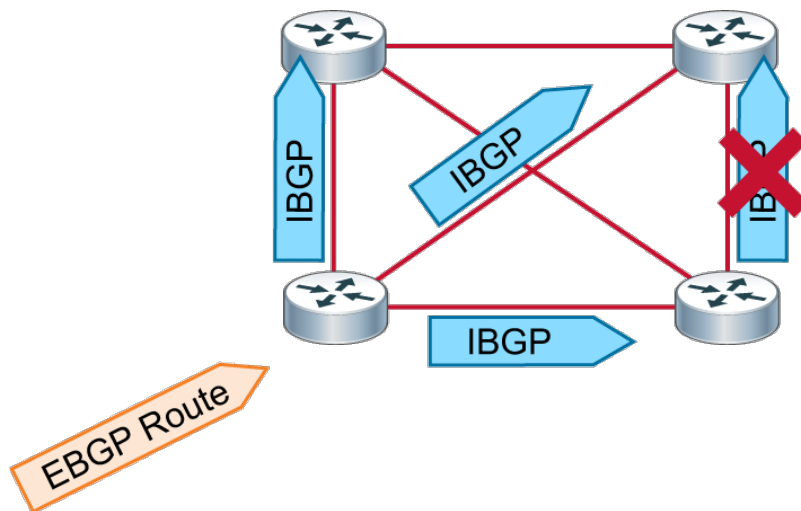
Route reflectors are BGP scalability mechanisms that enable routing information to be redistributed to all routers within an AS while eliminating the need for a fully meshed topology within the AS. This feature reduces the number of TCP sessions that must be maintained, lowering network overhead and [CPU](#) and memory resource requirements.

Two different solutions are available to achieve greater scalability when you are faced with the full-mesh rules of IBGP autonomous systems:

- Route reflectors modify the classic IBGP split-horizon rule and allow a particular router to forward incoming IBGP updates to an outgoing IBGP session under certain conditions. This router becomes a concentration router, or a route reflector.
- BGP confederations (not covered in this course) introduce the concept of a number of smaller autonomous systems within the original AS. The small autonomous systems exchange BGP updates between them using intra-confederation EBGP sessions.

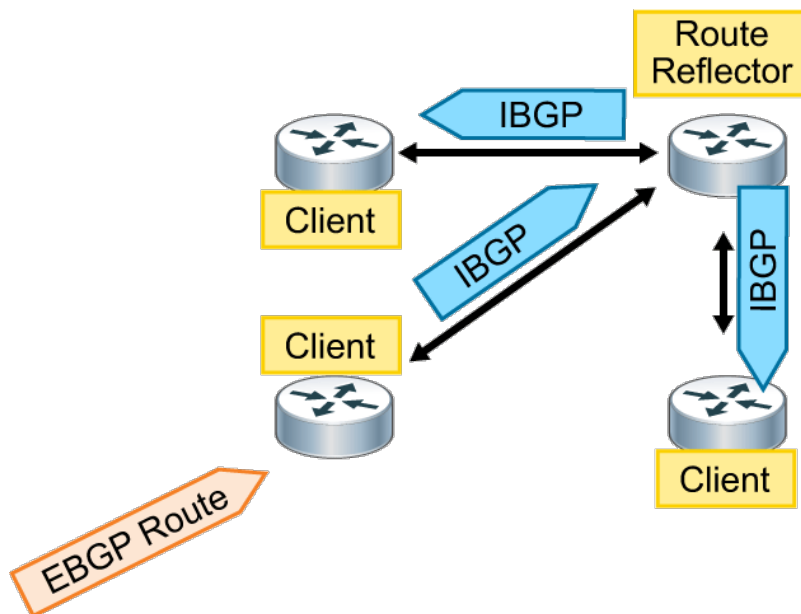
## Route Reflector Split-Horizon Rules

In classic **IBGP**, the **BGP** boundary router needs to forward the route that is received from an **EBGP** peer to every other router within its own **AS**. It uses a dedicated IBGP session for each router. Also, the BGP boundary router forwards routes that a router sources in the same way. To allow every router to update every other router, a full mesh of IBGP sessions is required.



- Classic IBGP: IBGP routes are not propagated to other IBGP peers.
- Full mesh of IBGP peers is therefore required.

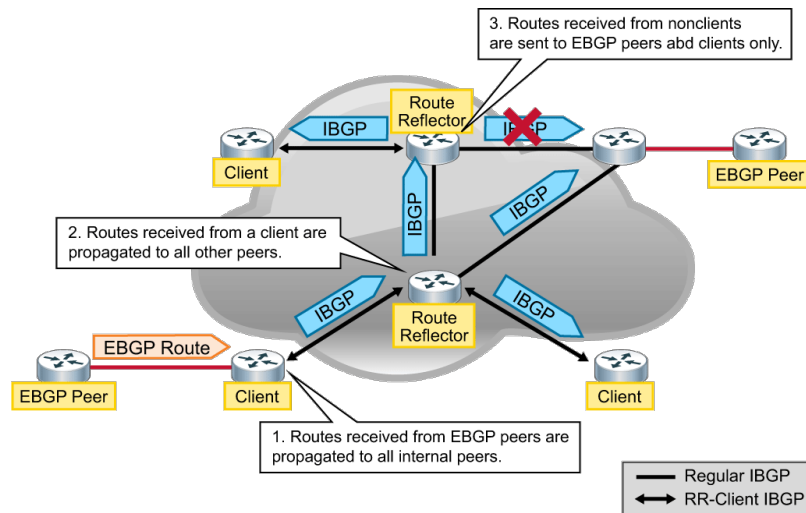
The figure shows how the router receives routing update on EBGP session and forwards this update to all IBGP neighbors. When IBGP neighbors receive this update on IBGP sessions, these updates are not forwarded to any other IBGP sessions by default.



- Route reflector can propagate IBGP routes to other IBGP peers.
- Full mesh of IBGP peers is no longer required.

The IBGP route reflector design relaxes the need for a full mesh. The router that is configured as a route reflector, under certain conditions, will relay updates that are received through an IBGP session to another IBGP session. This capability requires modifications of the classic IBGP split-horizon rules.

The route reflector concept introduces processing overhead on the concentration router and, if it is configured incorrectly, can cause routing loops and instability.

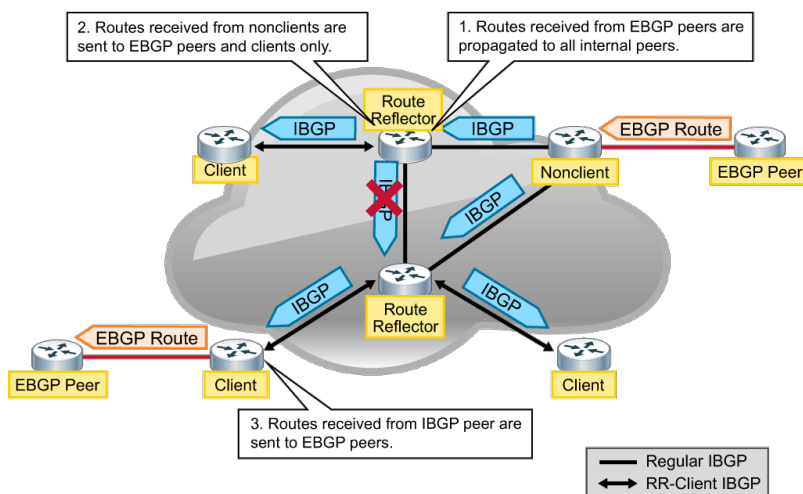


When you implement a route-reflector-based IBGP network, the BGP routers are divided into route reflectors (which implement modified split-horizon rules) and clients (which are behaving like traditional IBGP routers).

Route reflector clients are excluded from the full mesh. They can have any number of EBGP sessions but may have only one IBGP session, the session with their route reflector. Clients conform to the classic IBGP split-horizon rules and forward a received route from EBGP on their IBGP neighbor sessions. But the route reflector conforms to the route reflector split-horizon rules and recognizes that it has an IBGP session to a client. When the IBGP update is received from the client, the route reflector forwards the update to other IBGP neighbors, therefore alleviating the IBGP full-mesh requirement for its clients.

Similarly, when the route reflector receives an IBGP update from a neighbor that is not its client, it forwards the update to all of its clients.

Forwarding of an IBGP update in a route reflector does not change the next-hop attribute or any other common BGP attribute. This feature means that the client will use the optimum route with the recursive routing, regardless of the way that it has received the BGP route.



Another example of route propagation in a route reflector-enabled network:

- Nonclients conform to the classic IBGP split-horizon rules and forward a received route from EBGP on their IBGP neighbor sessions.
- The upper route reflector that receives a route from a nonclient sends the route to EBGP peers and clients only. It is the reason why the IBGP update is not sent to the bottom route reflector as well.
- The client receives the IBGP update and sends it to the EBGP peers.

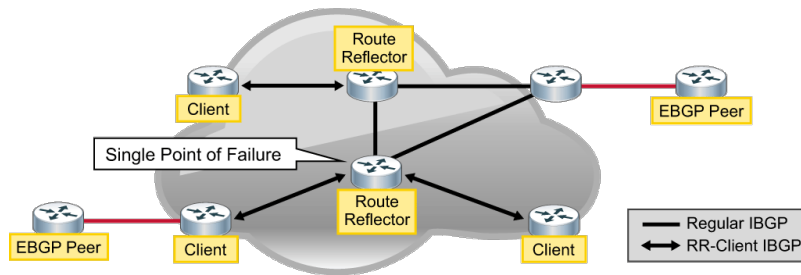
The table presents detailed IBGP split-horizon rules as modified by the introduction of BGP route reflectors. For purposes of definition, a "route reflector" is a BGP speaker that can advertise IBGP learned routes to another IBGP peer and, hence,

can reflect routes. IBGP peers of the route reflector fall under two categories: "clients" and "nonclients." The route reflector and its clients form a "cluster." All IBGP peers of the route reflector that are not part of the cluster are nonclients. A "classic" IBGP router is a router that does not support route reflector functionality.

| Type of Router  | Incoming Update From | Is Forwarded To           |
|-----------------|----------------------|---------------------------|
| Classic         | EBGP peer            | All peers (IBGP and EBGP) |
|                 | IBGP peer            | EBGP peers                |
| Route reflector | EBGP peer            | All peers (IBGP and EBGP) |
|                 | Nonclient IBGP peer  | EBGP peers and clients    |
|                 | Client IBGP peer     | All peers but the sender  |
| Client          | EBGP peer            | All peers (IBGP and EBGP) |
|                 | IBGP peer            | EBGP peers                |

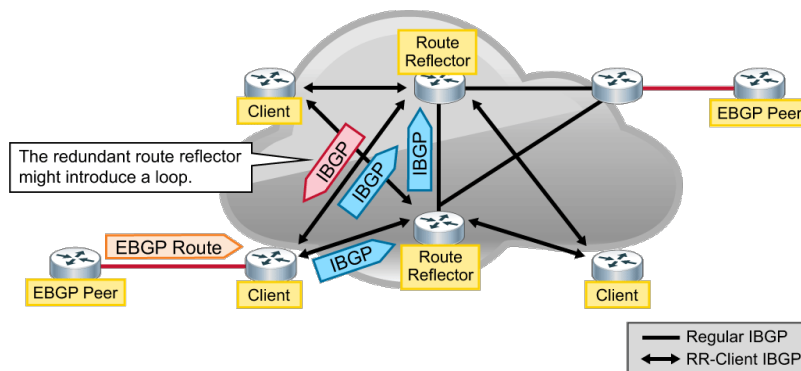
## Redundant Route Reflectors

Clients may have any number of **EBGP** peers but may have **IBGP** sessions only with their route reflector or reflectors. If the reflector fails, its client can no longer send **BGP** updates to, or receive them from, the rest of the **AS**. The route reflector is, therefore, a single point of failure.



- Clients that have an IBGP session with only one route reflector will not be able to send any BGP updates if the route reflector fails.
- Clients should establish an IBGP session with at least two route reflectors using different physical connections.

To avoid introducing a single point of failure into the network, the route reflector functionality must be as redundant as the physical network. If a client will still be physically attached to the network after its route reflector has failed, the client should have a redundant route reflector. Thus, in all highly available networks, route reflectors must be redundant.

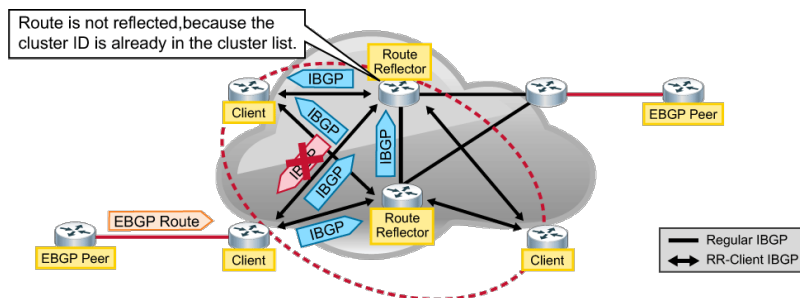


- Redundant reflectors solve the high-availability requirement.
- The concept of clusters is introduced to prevent IBGP routing loops between route reflectors.

A client may have IBGP sessions to more than one route reflector to avoid a single point of failure. Each client will receive the same route from both of its reflectors. Both route reflectors will receive the same IBGP update from their client, and they will both reflect the update to the rest of the clients. Additionally, both route reflectors will get updated from the full mesh and reflect those updates to their clients. As a result, each client will get two copies of all routes. Under certain circumstances (particularly when you use weights on IBGP sessions to influence BGP route selection), improper route reflection can result in an IBGP routing loop. And this routing loop is impossible to detect. Extra BGP attributes are thus necessary to prevent these routing loops.

## Route Reflector Clusters

A router that is acting as a route reflector client does not require any specific configuration. It simply has fewer [IBGP](#) sessions than it would have if it were part of the full mesh. But improperly configuring the client to also be a reflector could easily cause a loop. An IBGP route coming in from one of the real reflectors to the client could be forwarded by the client, erroneously acting as reflector, to the other reflector.



- A group of redundant route reflectors and their clients form a cluster.
- Each cluster must have a unique cluster ID.
- Each time a route is reflected, the cluster ID is added to the cluster-list BGP attribute.
- The route that already contains the local cluster ID in the cluster list is not reflected.

Route reflector clusters prevent IBGP routing loops in redundant route reflector designs.

The role of the network designer is to properly identify which route reflectors and their clients will form a cluster. The designer assigns to the cluster a cluster-ID number that is unique within the [AS](#).



The cluster-ID number must be configured in the route reflectors. The clients should not be configured with this information.

A route reflector router can reflect routes only within a single cluster. A route reflector can, however, participate in another cluster but only as a client. A client can function as a client only to a route reflector belonging to the same cluster.

When a route is reflected, the reflector creates the cluster-list attribute and attaches it to the route if it does not exist. It then sets its cluster-ID number in the cluster-list or adds its cluster-ID number to an already existing cluster-list attribute. If the route, for any reason, is ever reflected back to the same reflector, it will recognize its cluster-ID number in the cluster-list and not forward it again. The first route reflector that reflects the route also sets a [BGP](#) attribute, called "originator-ID," and adds it to the BGP router-ID of its client.



The cluster-list and originator-ID attributes are nontransitive optional BGP attributes, allowing routers that do not support route reflector functionality to coexist with route reflectors and their clients in the same AS.

Based on cluster-list and originator-ID attributes, routers can implement two loop-prevention mechanisms:

- Any router that receives an IBGP update with the originator-ID attribute set to its own BGP router-ID will ignore that update.
- Any route reflector that receives an IBGP update with its cluster-ID already in the cluster-list will ignore that update.

In the example, the client in the cluster forwards the received [EBGP](#) update to both reflectors. The route reflectors forward the update into the IBGP full mesh. This behavior means that they send the update to each other as well. But when a route reflector receives a BGP update from another route reflector, it recognizes their common cluster ID number in the cluster-list attribute. Therefore, the newly received route update is ignored.

## Additional Route Reflector Loop-Prevention Mechanisms

When a route is reflected, the route reflector sets the originator-ID [BGP](#) attribute (nontransitive optional BGP attribute) to the router-ID of the peer from which it received the route. Any router that receives a route with its own router-ID in the originator-ID attribute silently ignores that route.

- Every time a route is reflected, the router-ID of the originating IBGP router is stored in the originator-ID BGP attribute.
- A router receiving an IBGP route with originator-ID set to its own router-ID ignores that route.
- The BGP path selection procedure is modified to take into account cluster-list and originator-ID.

BGP path selection rules have been modified to select the best route in scenarios where a router might receive reflected and nonreflected routes or several reflected routes:

- The traditional BGP path selection parameters such as weight, local preference, origin, and [MED](#) are compared first.
- If these parameters are equal, the routes that are received from [EBGP](#) neighbors are preferred over routes that are received from [IBGP](#) neighbors.
- When a router receives two IBGP routes, the nonreflected routes (routes with no originator-ID attribute) are preferred over reflected routes.
- The reflected routes with shorter cluster-lists are preferred over routes with longer cluster-lists.
- If the additional route-reflector-oriented selection criteria do not yield a decision, the rest of the traditional BGP path selection rules are followed.

## Network Design with Route Reflectors

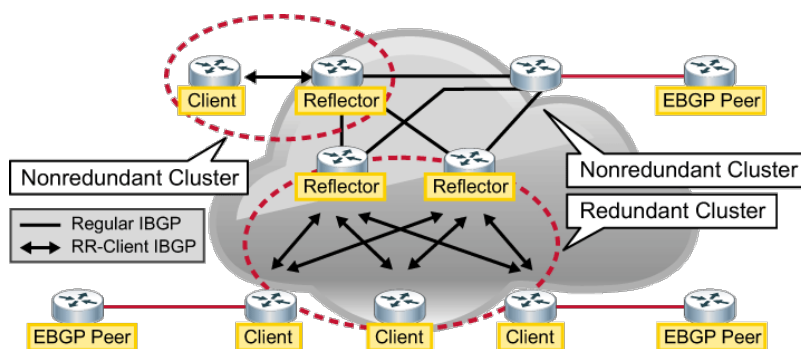
The physical topology of the network could serve as a guide to route reflector design.

Route reflector rules:

- Route reflector rules divide a transit AS into smaller areas (called clusters).
- Each cluster contains route reflectors and route reflector clients.
- Routers that do not support route reflector functionality act as a one-router cluster or as a route reflector client.

Implementing route reflectors within the transit AS will create smaller areas (or groups) of routers. These smaller groupings of routers are called clusters. A cluster consists of route reflector routers, either redundant or nonredundant, and the client routers that are connected to them.

In designing the implementation of route reflectors within a transit AS, identify a group of peripheral routers that are physically connected to the same backbone router or routers. Consider the peripheral routers as clients and the backbone routers as route reflectors. Then, consider this group of routers to form a cluster.



IBGP session rules:

- All clients in a cluster must establish IBGP sessions with and only with all route reflectors in the cluster.
- An IBGP full mesh between all route reflectors within the AS is required.
- Routers that are not route reflectors can participate in the IBGP full mesh or be route reflector clients.

The principal goal for designing networks with BGP route reflectors is to reduce the size of the full mesh of IBGP sessions by excluding some routers from the mesh. The routers that are excluded from the full mesh, the clients, have to send their IBGP information to, and receive it from, at least one router that belongs to the full mesh, the route reflector. Thus, the full mesh is still there, but it is smaller, and all route reflectors have to be part of it.

All clients in a cluster should have IBGP sessions with all their route reflectors and their route reflectors only. If a client does not have sessions with all the reflectors in the cluster, the redundancy is violated. If a client has IBGP sessions to routers other than the route reflectors, unnecessary routing traffic is generated.

Both clients and other routers, that are not route reflectors, obey the classic IBGP split-horizon rules. Thus, non-route-reflector routers are either clients to a reflector or are participating directly in the full mesh.

In the example, there are two clusters and one standalone BGP routers. In the area that is labeled "redundant cluster," the three client routers and the two route reflector routers make up the cluster. Each of the three client routers has an IBGP session with the two route reflectors and only with those two route reflectors.

In the nonredundant area, the client router has a single physical connection to a route reflector router. These two routers form a nonredundant cluster. The router that is designated as the route reflector in the cluster is already a single point of failure in this physical design. A failure of this router will prevent the clients in the cluster from reaching the rest of the network. Therefore, there is no new single point of failure that is introduced when the router is configured as the only route reflector in this cluster. The client has a single IBGP session to the route reflector.

The standalone router that is shown is not configured as a route reflector, nor is it a



client to any other route reflector. This other router serves as an example of where a non-route-reflector router participates in the full mesh. This router has a full mesh of IBGP sessions with all route reflectors, each reflector representing its cluster.

## Potential Network Issues

Potential problems that can occur when you deviate from the route reflector network design rules:

| Issue:                                                         | Result:                                                  |
|----------------------------------------------------------------|----------------------------------------------------------|
| Clients do not have sessions with all reflectors in a cluster. | Clients will not receive all IBGP routes.                |
| Clients have sessions with reflectors in several clusters.     | Clients will receive duplicate copies of the same route. |
| Clients have IBGP sessions with other clients.                 | Clients will receive duplicate copies of the same route. |

Two nontransitive optional [BGP](#) attributes, originator-ID and cluster-list, are both used to prevent fatal loops of information. The use of these two attributes makes a network fairly insensitive to poor configuration. However, for optimal performance, you must have an optimal configuration. Here are some of the problems that could occur if you deviate from route reflector network design rules:

- If route reflectors are not connected with [IBGP](#) sessions in a full mesh, some clusters will not have all the routes.
- If a client has IBGP sessions with some route reflectors in a cluster, but not with all of them, the client might miss some BGP routes.
- If a client has IBGP sessions to route reflectors that belong to different clusters, the client will forward the BGP update from the client into the full mesh with different cluster-IDs in the cluster-list attribute. When the BGP update enters the mesh, it will reach the other route reflector, which will, unnecessarily, accept the route as valid and forward it into its cluster. This situation, in turn, causes unnecessary duplication of updates to the clients.
- If a client has IBGP sessions to other clients in the same cluster, those clients will receive unnecessary duplications of updates.

## Hierarchical Route Reflectors

Network designers can build route reflector clusters in hierarchies. With hierarchies, a router serving as a route reflector in one cluster can act as a client in another cluster.

### Problem:

- In very large networks, a single layer of route reflectors might not be enough.

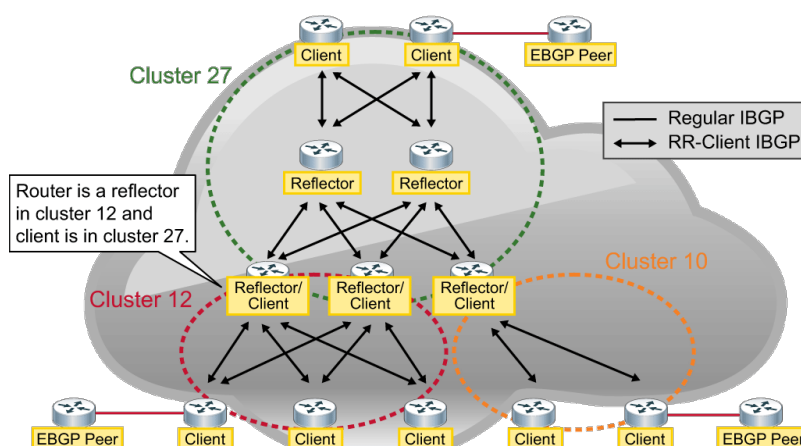
### Solution:

- A hierarchy of route reflectors can be established.
  - A route reflector can be a client of another route reflector.
  - The hierarchy can be as deep as needed.

Clients are not configured to be route reflector clients; they simply have fewer [IBGP](#) sessions. However, a network designer must configure a route reflector. In configuring an IBGP session on a route reflector, the designer must configure the session to reach a client, so the route reflector IBGP split-horizon rules to start working. All other IBGP sessions that are configured on the route reflector are a part of the full mesh. Also, the designer must configure the cluster-ID on the route reflector.

A router that is configured to be a route reflector will still have ordinary IBGP sessions that are part of the full mesh. If these sessions are reduced in number and only a few remain, and the remaining ones reach a second level of route reflectors, a hierarchy of route reflectors is created.

When a designer builds a first level of clusters, the remaining full mesh is smaller than when all routers belonged to it. But if it is large enough, the designer can build an extra level of route reflectors.



The figure shows an example of hierarchical route reflectors.

The first level of hierarchy reduced the original full mesh of 12 routers (all routers in the service provider network) to a full mesh of seven routers (the lower three route reflectors and the upper two route reflectors and two clients). The second level of route reflector clusters was built by creating cluster 27. This second step further reduced the full mesh of seven routers to a full mesh consisting of only two routers (upper route reflectors). Only the two route reflectors in cluster 27 should be connected in a full mesh.

When a client in the lowest level receives an [EBGP](#) update, it will forward the update on all configured IBGP sessions to a route reflector. The route reflector recognizes [BGP](#) updates that are received from configured clients and will forward these updates to all other clients that use normal IBGP sessions. The update, sent on a normal IBGP session, will be a second-level client update to the second-level route reflector. The second-level route reflector will recognize that the update was received from a client, and will forward it to all other clients and into the full mesh.

## Summary

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

- BGP route reflectors were introduced to free the network designers from IBGP full-mesh requirements that prevent large networks from scaling.
- BGP route reflectors modify IBGP split-horizon rules. All routes that are received from a route reflector client are sent to all other IBGP neighbors. All routes that are received from a nonclient IBGP neighbor are sent to all route reflector clients.
- A route reflector is a single point of failure, and therefore redundancy should be implemented in a network containing route reflectors.
- Route reflector clusters were introduced in the BGP route reflector architecture to support redundancy, preventing IBGP routing loops in redundant route reflector designs.
- The originator-ID and cluster-list BGP attributes were introduced to prevent routing loops in route reflector environments.
- All route reflectors in a cluster should have IBGP sessions to all clients in the cluster. The route reflectors also participate in the IBGP full mesh, and they should have no other IBGP sessions.
- When the route reflector clients do not have IBGP sessions with all route reflectors in the cluster, they might not receive all IBGP routes.
- Route reflector clusters can be built in hierarchies. A router that is a route reflector in one cluster can act as client in another cluster.

# Configuring and Monitoring Route Reflectors

## Overview

Large [BGP](#) networks cannot properly scale without relying on performance-enhancing tools such as route reflectors. Route reflectors enable BGP routing information to be distributed in a fashion that does not require a physical full-mesh network. Implementing such a network requires knowledge of the steps to properly migrate and configure route reflectors and the commands that are used to verify the operation of a configured network.

This lesson introduces the steps that are required to successfully migrate an existing [AS](#) to BGP route reflectors. It also lists the Cisco IOS commands that are required to configure and monitor route reflectors.

Upon completing this lesson, you will be able to:

- List the steps to migrate an existing IBGP backbone to a backbone with route reflectors
- Identify the configuration changes and related Cisco IOS commands that are required to configure route reflectors on a BGP backbone
- Identify the Cisco IOS commands that are required to monitor a BGP backbone that contains route reflectors

## Route Reflector Backbone Migration

The physical topology of the [AS](#) serves as a guide to designing clusters. You should introduce no additional single points of failure when you are deploying route reflectors. If the physical topology is redundant, a good practice is to have redundant route reflectors. If the physical topology is not redundant, introducing a nonredundant cluster does not add a single point of failure because the network was already nonredundant.

- Divide the AS into areas (clusters).
  - Assign a cluster-ID to each area.
- On route reflector clients, retain only IBGP sessions with route reflectors in their cluster.
- On route reflectors, retain only IBGP sessions with other route reflectors and clients in their cluster.
  - Configure cluster-ID on every route reflector.
  - Configure clients on every route reflector.

The following planning and preparation steps are required before you migrate from a full mesh of [IBGP](#) sessions to a route reflector design:

1. Identify a group of peripheral routers that are physically connected to the same set of backbone routers. Consider the peripheral routers as clients and the backbone routers as route reflectors. Let the routers form a cluster. Make sure that no router belongs to two different clusters, because this setup would represent an illegal configuration.
2. Create a numbering plan that indicates how numbers are assigned to the clusters in the network. The plan must make sure to uniquely identify each of the clusters within the AS. Clusters are not seen from outside the AS, so the plan does not need to be coordinated with any other AS. To ease troubleshooting, it is recommended that numbers lower than 256 are used, because cluster-IDs are displayed in [IP address](#) format.



The default value of a cluster-ID is the [BGP](#) router-ID of the route reflector. If you decide to implement nonredundant clusters, you do not have to plan the cluster-ID numbers, because the BGP router-IDs should be unique.

## Configuring Route Reflectors

Configuration changes that are required to configure BGP route reflectors:

- Configure cluster-ID on route reflectors.
- Configure BGP neighbors as route reflector clients on the route reflectors.
- No configuration is needed on the route reflector clients.
- Make sure that IBGP neighbor is removed on both ends of the IBGP session.

As part of the planning and preparation that is necessary to migrate from a full mesh of [IBGP](#) sessions to a route reflector design, you need to make the following configuration changes:

- Configure the proper cluster-ID value on the route reflectors.
- Configure the route reflector with information about which IBGP neighbor sessions are reaching their clients.
- In the clients, remove all IBGP sessions to neighbors that are not route reflectors in the client cluster.
- Make sure that the IBGP neighbor is removed on both ends of the IBGP session.

Cisco IOS commands that are required to configure BGP route reflectors:

```
router(config-router)# bgp cluster-id cluster-id
```

- Optionally assigns a cluster-ID to the route reflector (default value is router-ID)
- Required only for clusters with redundant reflectors

```
router(config-router)# neighbor ip-address route-reflector-client
```

- On route reflector configures an IBGP neighbor to be a client of this reflector.

Use the **bgp cluster-id** command to configure the cluster-ID if the [BGP](#) cluster has redundant route reflectors.

```
bgp cluster-id cluster-id
```

To remove the cluster-ID, use the **no** form of this command.

```
no bgp cluster-id cluster-id
```

### Syntax Description

| Parameter         | Description                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------|
| <i>cluster-id</i> | Cluster-ID of the router acting as a route reflector.<br>The cluster-ID is a maximum of 4 bytes. |

This command is used to configure the router as a BGP route reflector and configure the specified neighbor as its client. When all the clients are disabled, the local router is no longer a route reflector.

```
neighbor ip-address route-reflector-client
```

To indicate that the neighbor is not a client, use the **no** form of this command.

```
no neighbor ip-address route-reflector-client
```

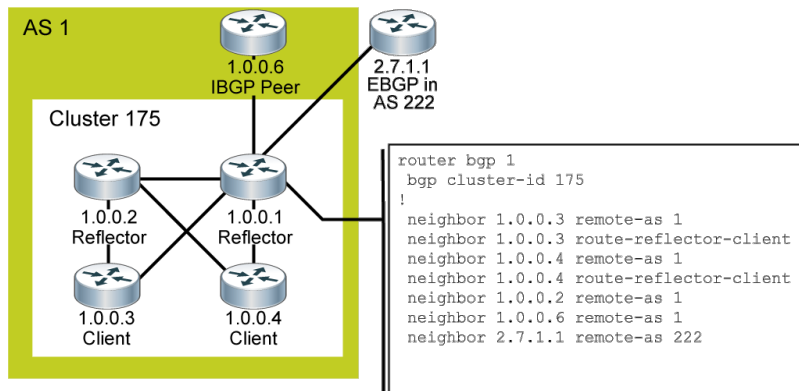
### Syntax Description

| Parameter         | Description         |
|-------------------|---------------------|
| <i>ip-address</i> | Neighbor IP address |

By default, there is no route reflector in the [AS](#).

## Configuring Route Reflectors Example

In this example, the router 1.0.0.1 has been configured as a route reflector.



The routers with router-ID 1.0.0.1 and 1.0.0.2 are route reflectors in a cluster that has been assigned cluster-ID 175. The routers with router-ID 1.0.0.3 and 1.0.0.4 are clients to these two route reflectors.

The figure shows a portion of the configuration in router 1.0.0.1. The cluster-ID is assigned to the router under the **router bgp** process definition of the router configuration. After the router has been assigned, the route reflector client configuration is added under the **router bgp** process for the two neighbors that identify the two sessions reaching clients.

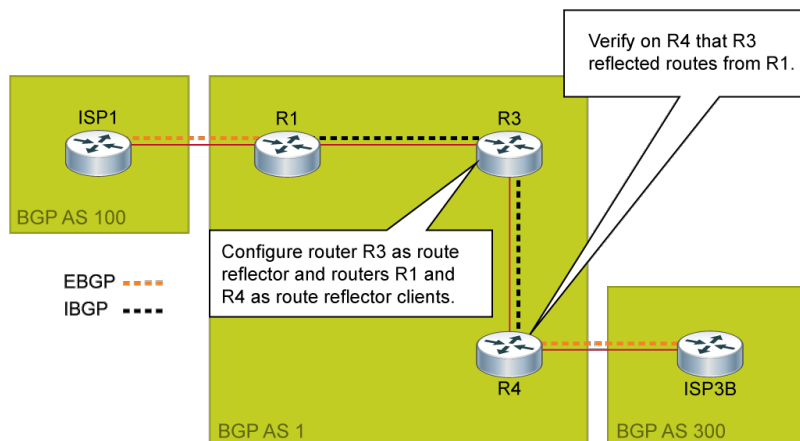
Both clients need to have established IBGP session to both route reflectors. IBGP session between route reflectors is also needed. Route reflector 1.0.0.1 also has established IBGP session to nonclient router 1.0.0.6 and **EBGP** session to router 2.7.1.1 on autonomous system 222.



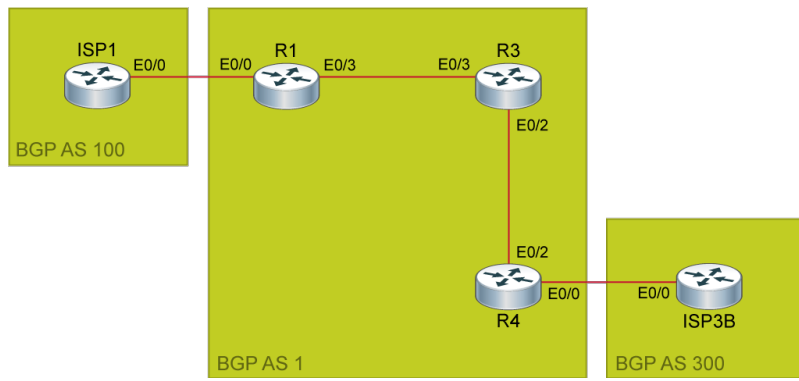
## Discovery 17: Configure Route Reflector

### Overview

Through this discovery, you will learn how to eliminate the need for full mesh [IBGP](#) sessions by introducing the route reflector feature into the autonomous system.



## Topology



## Job Aids

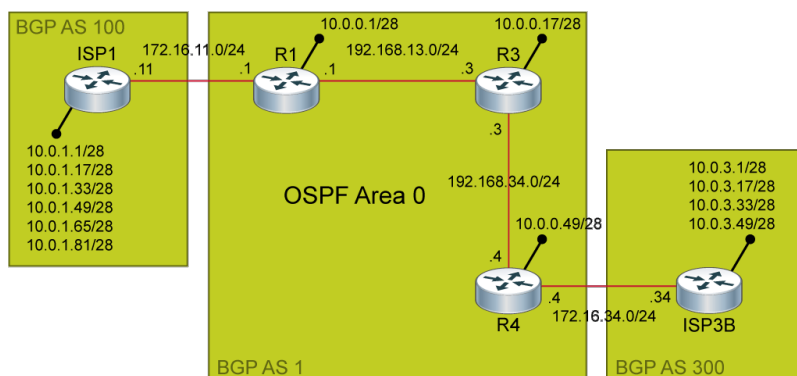


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

### Device Information

| Device | Interface                                                                        | IP address                                                                                  | Description                     |
|--------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------|
| ISP1   | Ethernet 0/0                                                                     | 172.16.11.11/24                                                                             | Connection to R1                |
| ISP1   | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4<br>Loopback 5<br>Loopback 6 | 10.0.1.1/28<br>10.0.1.17/28<br>10.0.1.33/28<br>10.0.1.49/28<br>10.0.1.65/28<br>10.0.1.81/28 | Loopbacks simulate LAN networks |
| R1     | Ethernet 0/0                                                                     | 172.16.11.1/24                                                                              | Connection to ISP1              |
| R1     | Ethernet 0/3                                                                     | 192.168.13.1/24                                                                             | Connection to R3                |
| R1     | Loopback 1                                                                       | 10.0.0.1/28                                                                                 | Loopback simulates LAN network  |
| R3     | Ethernet 0/2                                                                     | 192.168.34.3/24                                                                             | Connection to R4                |
| R3     | Ethernet 0/3                                                                     | 192.168.13.3/24                                                                             | Connection to R1                |
| R3     | Loopback 1                                                                       | 10.0.0.17/28                                                                                | Loopback simulates LAN network  |
| R4     | Ethernet 0/0                                                                     | 172.16.34.4/24                                                                              | Connection to ISP3B             |
| R4     | Ethernet 0/3                                                                     | 192.168.24.4/24                                                                             | Connection to R2                |
| R4     | Loopback 1                                                                       | 10.0.0.49/28                                                                                | Loopback simulates LAN network  |
| ISP3B  | Ethernet 0/0                                                                     | 172.16.34.34/24                                                                             | Connection to R4                |
| ISP3B  | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4                             | 10.0.3.1/28<br>10.0.3.17/28<br>10.0.3.33/28<br>10.0.3.49/28                                 | Loopbacks simulate LAN networks |

IP addresses and IGP are preconfigured as shown in the topology below:



These BGP sessions have been preconfigured:

- EBGp session between ISP1 and R1 routers
- EBGp session between ISP3B and R4 routers
- IBGP session between R1 and R3 routers
- IBGP session between R3 and R4 routers

Note that there is no IBGP session between R1 and R4 routers. OSPF protocol has also been preconfigured as IGP routing protocol, making sure that all internal links,

loopbacks, and next hops are accessible in autonomous system AS 1.

# Configuring Route Reflector

## Discovery Steps

### Step 1

On the R1 router, verify which BGP sessions are established.

```
R1# show bgp summary
BGP router identifier 10.0.0.1, local AS number 1
BGP table version is 101, main routing table version 101
8 network entries using 1184 bytes of memory
8 path entries using 512 bytes of memory
4/3 BGP path/bestpath attribute entries using 544 bytes of memory
1 BGP rrinfo entries using 24 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 2312 total bytes of memory
BGP activity 55/42 prefixes, 86/78 paths, scan interval 60 secs
```

| Neighbor     | V | AS  | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down  | State/PfxRcd |
|--------------|---|-----|---------|---------|--------|-----|------|----------|--------------|
| 10.0.0.17    | 4 | 1   | 8       | 6       | 101    | 0   | 0    | 00:00:22 | 1            |
| 172.16.11.11 | 4 | 100 | 109     | 118     | 101    | 0   | 0    | 01:33:46 | 6            |

You should only see the BGP sessions established:

- EBGP session to the ISP1 router (neighbor 172.16.11.11)
- IBGP session to the R3 router (neighbor 10.0.0.17)

### Step 2

On the R3 router, verify which BGP sessions are established.

```
R3# show bgp summary
BGP router identifier 10.0.0.17, local AS number 1
BGP table version is 198, main routing table version 198
13 network entries using 1924 bytes of memory
13 path entries using 832 bytes of memory
4/4 BGP path/bestpath attribute entries using 544 bytes of memory
2 BGP AS-PATH entries using 48 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3348 total bytes of memory
BGP activity 37/24 prefixes, 109/96 paths, scan interval 60 secs
```

| Neighbor  | V | AS | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down  | State/PfxRcd |
|-----------|---|----|---------|---------|--------|-----|------|----------|--------------|
| 10.0.0.1  | 4 | 1  | 7       | 8       | 198    | 0   | 0    | 00:00:51 | 7            |
| 10.0.0.49 | 4 | 1  | 6       | 7       | 198    | 0   | 0    | 00:00:38 | 5            |

You should only see the BGP sessions established:

- IBGP session to the R1 router (neighbor 10.0.0.1)
- IBGP session to the R4 router (neighbor 10.0.0.49)

### Step 3

On the R4 router, verify which BGP sessions are established.

```
R4# show bgp summary
BGP router identifier 10.0.0.49, local AS number 1
BGP table version is 145, main routing table version 145
6 network entries using 888 bytes of memory
6 path entries using 384 bytes of memory
3/3 BGP path/bestpath attribute entries using 408 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1704 total bytes of memory
BGP activity 61/55 prefixes, 109/103 paths, scan interval 60 secs
```

| Neighbor     | V | AS  | MsgRcvd | MsgSent | TblVer | InQ | OutQ | Up/Down  | State/PfxRcd |
|--------------|---|-----|---------|---------|--------|-----|------|----------|--------------|
| 10.0.0.17    | 4 | 1   | 8       | 7       | 145    | 0   | 0    | 00:01:03 | 1            |
| 172.16.34.34 | 4 | 300 | 109     | 122     | 145    | 0   | 0    | 01:34:45 | 4            |

You should only see the BGP sessions established:

- EBGP session to the ISP3B router (neighbor 172.16.34.34)
- IBGP session to the R3 router (neighbor 10.0.0.17)

**Note** that there is no IBGP session between routers R1 and R4.

#### Step 4

On the R1 router, verify which routes are advertised to the router R3.

```
R1# show ip bgp neighbors 10.0.0.17 advertised-routes
BGP table version is 53, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|    | Network      | Next Hop     | Metric | LocPrf | Weight | Path  |
|----|--------------|--------------|--------|--------|--------|-------|
| *> | 10.0.0.0/28  | 0.0.0.0      | 0      |        | 32768  | i     |
| *> | 10.0.1.0/28  | 172.16.11.11 | 0      |        | 0      | 100 i |
| *> | 10.0.1.16/28 | 172.16.11.11 | 0      |        | 0      | 100 i |
| *> | 10.0.1.32/28 | 172.16.11.11 | 0      |        | 0      | 100 i |
| *> | 10.0.1.48/28 | 172.16.11.11 | 0      |        | 0      | 100 i |
| *> | 10.0.1.64/28 | 172.16.11.11 | 0      |        | 0      | 100 i |
| *> | 10.0.1.80/28 | 172.16.11.11 | 0      |        | 0      | 100 i |

You should see that R1 router advertises these prefixes:

- Locally originated prefix 10.0.0.0/28
- All prefixes originating on the ISP1 router (10.0.1.0/24 subnets) in AS 100.

#### Step 5

On the R3 router, verify the content of the BGP table.

```
R3# show ip bgp
BGP table version is 102, local router ID is 10.0.0.17
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|     | Network      | Next Hop  | Metric | LocPrf | Weight | Path  |
|-----|--------------|-----------|--------|--------|--------|-------|
| *>i | 10.0.0.0/28  | 10.0.0.1  | 0      | 100    | 0      | i     |
| *>  | 10.0.0.16/28 | 0.0.0.0   | 0      |        | 32768  | i     |
| *>i | 10.0.0.48/28 | 10.0.0.49 | 0      | 100    | 0      | i     |
| *>i | 10.0.1.0/28  | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.1.16/28 | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.1.32/28 | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.1.48/28 | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.1.64/28 | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.1.80/28 | 10.0.0.1  | 0      | 100    | 0      | 100 i |
| *>i | 10.0.3.0/28  | 10.0.0.49 | 0      | 100    | 0      | 300 i |
| *>i | 10.0.3.16/28 | 10.0.0.49 | 0      | 100    | 0      | 300 i |
| *>i | 10.0.3.32/28 | 10.0.0.49 | 0      | 100    | 0      | 300 i |
| *>i | 10.0.3.48/28 | 10.0.0.49 | 0      | 100    | 0      | 300 i |

You should see that prefixes originating on the ISP1 router (10.0.1.0/24 prefixes) were received on the R3 router.

#### Step 6

On the R3 router, verify which routes are advertised to the router R4.

```
R3# show ip bgp neighbors 10.0.0.49 advertised-routes
BGP table version is 102, local router ID is 10.0.0.17
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

|    | Network      | Next Hop | Metric | LocPrf | Weight | Path |
|----|--------------|----------|--------|--------|--------|------|
| *> | 10.0.0.16/28 | 0.0.0.0  | 0      |        | 32768  | i    |

```
Total number of prefixes 1
```

You should see that only locally originated prefix (10.0.0.16/28) is advertised to the R4 router. Split horizon rule prevents router R3 to pass received updates from router R1 to be sent to the R4 router.

### Step 7

On the R4 router, verify the content of the BGP table.

```
R4# show ip bgp
BGP table version is 81, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*>i 10.0.0.16/28 10.0.0.17 0 100 0 i
*> 10.0.0.48/28 0.0.0.0 0 32768 i
*> 10.0.3.0/28 172.16.34.34 0 0 300 i
*> 10.0.3.16/28 172.16.34.34 0 0 300 i
*> 10.0.3.32/28 172.16.34.34 0 0 300 i
*> 10.0.3.48/28 172.16.34.34 0 0 300 i
```

You should see that prefixes originating on the ISP1 router (10.0.1.0/24 prefixes) were not received on the R4 router.

**NOTE:** For the prefixes originating on the ISP1 router to be received on the R4 router, either IBGP session between R1 and R4 needs to be configured or the router R3 should be configured as the route reflector to break the split horizon rule for IBGP sessions.

### Step 8

On the ISP3B router, verify the content of the BGP table.

```
ISP3B# show ip bgp
BGP table version is 75, local router ID is 10.0.3.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.16/28 172.16.34.4 0 0 1 i
*> 10.0.0.48/28 172.16.34.4 0 0 1 i
*> 10.0.3.0/28 0.0.0.0 0 32768 i
*> 10.0.3.16/28 0.0.0.0 0 32768 i
*> 10.0.3.32/28 0.0.0.0 0 32768 i
*> 10.0.3.48/28 0.0.0.0 0 32768 i
```

You should see as a consequence of the IBGP split horizon rule that prefixes originating on the ISP1 router (10.0.1.0/24 prefixes) were not received on the ISP3B router.

### Step 9

Configure the BGP cluster ID of 1 on the R3 router. Configure the R3 router to be route reflector for client router R1 and R4.

```
R3(config)# router bgp 1
R3(config-router)# bgp cluster-id 1
R3(config-router)# neighbor 10.0.0.1 route-reflector-client
R3(config-router)# neighbor 10.0.0.49 route-reflector-client
```

**NOTE:** IBGP session is restored automatically after route reflector client is configured on the route reflector router.

## Monitoring Route Reflectors

Commands that are used to monitor route reflector operation:

```
router# show ip bgp neighbors
```

- Displays whether a neighbor is a route reflector client.

```
router# show ip bgp network/lenght
```

- Displays the routes that are received from the client as seen on the reflector.
- Displays reflected routes as seen on the client (with originator ID and cluster ID information)

To display information about the TCP and BGP connections to neighbors, use the **show ip bgp neighbors EXEC** command.

```
show ip bgp neighbors [address]
```

In this case, the **show ip bgp neighbors** command is used on the router not to see routes or paths that have been received but to see the status of the neighbor session, so no other qualifiers than the optional IP address are given. The command that is issued on the route reflector router, indicates that the neighbor is a route reflector client.

To display entries in the BGP routing table, use the **show ip bgp EXEC** command.

```
show ip bgp[network/lenght [longer-prefixes]
```

When details are displayed for a specific route entry in the BGP table, the cluster list and originator ID attributes are also shown. If you issue this command on the route reflector, you could see that a particular entry was received from a route reflector client. If you issue this command on the route reflector client, you could see entries in the BGP table, that were reflected by the route reflector.

### Step 10

On the R3 router (which acts as a route reflector), verify that the R1 router is configured as route reflector client.

```
R3# show ip bgp neighbors 10.0.0.1
BGP neighbor is 10.0.0.1, remote AS 1, internal link
 BGP version 4, remote router ID 10.0.0.1
 BGP state = Established, up for 00:07:39
 Last read 00:00:41, last write 00:00:22, hold time is 180, keepalive interval is 60 seconds
 Neighbor sessions:
 1 active, is not multisession capable (disabled)
 Neighbor capabilities:
 Route refresh: advertised and received(new)
 Four-octets ASN Capability: advertised and received
 Address family IPv4 Unicast: advertised and received
 Enhanced Refresh Capability: advertised and received
 Multisession Capability:
 Stateful switchover support enabled: NO for session 1
 Message statistics:
 InQ depth is 0
 OutQ depth is 0

 Sent Rcvd
 Opens: 1 1
 Notifications: 0 0
 Updates: 7 3
 Keepalives: 10 10
 Route Refresh: 0 0
 Total: 18 14
 Default minimum time between advertisement runs is 0 seconds

For address family: IPv4 Unicast
 Session: 10.0.0.1
 BGP table version 150, neighbor version 150/0
 Output queue size : 0
 Index 5, Advertise bit 1
 Route-Reflector Client
 5 update-group member
```



```
Slow-peer detection is disabled
Slow-peer split-update-group dynamic is disabled
```

### Step 11

On the R3 router (which acts as a route reflector), verify additional information in the BGP table for prefix 10.0.1.0/28, which is originated on the ISP1 router. Make sure that the route was received from the route reflector client.

```
R3# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 134
Paths: (1 available, best #1, table default)
 Advertised to update-groups:
 5
 Refresh Epoch 1
 100, (Received from a RR-client)
 10.0.0.1 (metric 11) from 10.0.0.1 (10.0.0.1)
 Origin IGP, metric 0, localpref 100, valid, internal, best
```

### Step 12

On the R4 router (which acts as a route reflector client), verify additional information in the BGP table for prefix 10.0.1.0/28, which is originated on the ISP1 router. Make sure that the route was reflected from the route reflector router R3.

```
R4# show ip bgp 10.0.1.0/28
BGP routing table entry for 10.0.1.0/28, version 99
Paths: (1 available, best #1, table default)
 Advertised to update-groups:
 6
 Refresh Epoch 2
 100
 10.0.0.1 (metric 21) from 10.0.0.17 (10.0.0.17)
 Origin IGP, metric 0, localpref 100, valid, internal, best
 Originator: 10.0.0.1, Cluster list: 0.0.0.1
```

You should see that originator of this reflected route was route reflector client R1 (10.0.0.1) and the BGP cluster ID is 1.

### Step 13

On the R3 router, verify which routes are advertised to the R4 router.

```
R3# show ip bgp neighbors 10.0.0.49 advertised-routes
BGP table version is 150, local router ID is 10.0.0.17
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*>i 10.0.0.0/28 10.0.0.1 0 100 0 i
*> 10.0.0.16/28 0.0.0.0 0 32768 i
*>i 10.0.0.48/28 10.0.0.49 0 100 0 i
*>i 10.0.1.0/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.16/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.32/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.48/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.64/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.80/28 10.0.0.1 0 100 0 100 i
*>i 10.0.3.0/28 10.0.0.49 0 100 0 300 i
*>i 10.0.3.16/28 10.0.0.49 0 100 0 300 i
*>i 10.0.3.32/28 10.0.0.49 0 100 0 300 i
*>i 10.0.3.48/28 10.0.0.49 0 100 0 300 i

 Network Next Hop Metric LocPrf Weight Path
Total number of prefixes 13
```

You should see that prefixes originated on the ISP1 (10.0.1.0/24 subnets) router are being advertised from the R3 router to the IBGP neighbor R4 because of the router R3 acting as a route reflector.

#### Step 14

On the R4 router, verify the content of the BGP table.

```
R4# show ip bgp
BGP table version is 106, local router ID is 10.0.0.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*>i 10.0.0.0/28 10.0.0.1 0 100 0 i
*>i 10.0.0.16/28 10.0.0.17 0 100 0 i
*> 10.0.0.48/28 0.0.0.0 0 32768 i
*>i 10.0.1.0/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.16/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.32/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.48/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.64/28 10.0.0.1 0 100 0 100 i
*>i 10.0.1.80/28 10.0.0.1 0 100 0 100 i
*> 10.0.3.0/28 172.16.34.34 0 0 300 i
*> 10.0.3.16/28 172.16.34.34 0 0 300 i
*> 10.0.3.32/28 172.16.34.34 0 0 300 i
*> 10.0.3.48/28 172.16.34.34 0 0 300 i
```

You should see that prefixes originating on the ISP1 router (10.0.1.0/24 prefixes) were received on the R4 router, because the router R3 is configured as a route reflector.

#### Step 15

On the ISP3B router, verify the content of the BGP table.

```
ISP3B# show ip bgp
BGP table version is 84, local router ID is 10.0.3.49
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/28 172.16.34.4 0 0 1 i
*> 10.0.0.16/28 172.16.34.4 0 0 1 i
*> 10.0.0.48/28 172.16.34.4 0 0 1 i
*> 10.0.1.0/28 172.16.34.4 0 0 1 100 i
*> 10.0.1.16/28 172.16.34.4 0 0 1 100 i
*> 10.0.1.32/28 172.16.34.4 0 0 1 100 i
*> 10.0.1.48/28 172.16.34.4 0 0 1 100 i
*> 10.0.1.64/28 172.16.34.4 0 0 1 100 i
*> 10.0.1.80/28 172.16.34.4 0 0 1 100 i
*> 10.0.3.0/28 0.0.0.0 0 32768 i
*> 10.0.3.16/28 0.0.0.0 0 32768 i
*> 10.0.3.32/28 0.0.0.0 0 32768 i
*> 10.0.3.48/28 0.0.0.0 0 32768 i
```

You should see that prefixes originating on the ISP1 router (10.0.1.0/24 prefixes) were received on the ISP3B router.

## Summary

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

- A proper migration plan is important because the change to BGP confederation involves a major reconfiguration of BGP routing.
- The **neighbor *ip-address* route-reflector-client** command specifies an IBGP neighbor to be a client of this reflector
- The **show ip bgp neighbors** command has been modified to display whether a BGP neighbor is part of a BGP confederation.

# Module Summary

## Overview

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

- Service providers use an IGP to carry internal routes and to provide optimal routing between POPs, the information that is needed for IBGP sessions to be established, and the addresses that are required for BGP next-hop resolution.
- Route reflectors enable BGP routing information to be distributed in a fashion that does not require a physical fully meshed network.
- Route reflector clusters can be built in hierarchies. A router that is a route reflector in one cluster can act as a client in another cluster.
- There are only two Cisco IOS commands that are used to configure route reflectors: **bgp cluster-id** and neighbor *ip-address* route-reflector-client.

## 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 three characteristics are common to typical service provider networks? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

- ☐ The provider network uses two IGPs, one for customer routes and one for internal provider routes.
- ☐ Service providers exchange routes with other providers using BGP.
- ☐ Service providers run IBGP within their network in addition to their IGP requirements.
- ☐ Service providers typically use either static routes or EBGP with their customers.

1. What is the typical role of an IGP within a service provider network?  
(Source: "Scaling IGP and BGP in Service Provider Networks")

• The IGP carries customer routes for redistribution into BGP at the provider edge.

• The IGP advertises a default route to customers of the service provider.

• The IGP resolves next-hop IP addresses.

• The IGP carries BGP routes across the provider network.

1. The IGP protocol is not needed in the service provider network, as information about next hop attribute gets advertised via BGP updates. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ true

☐ false



1. Why should you avoid the use of private IP addressing in service provider networks? (Source: "Scaling IGP and BGP in Service Provider Networks")

- Private addressing can prevent customer network troubleshooting utilities such as traceroute from functioning correctly.

- Private IP addressing is not allowed on the Internet and will not function in a service provider network.

- Private IP addressing prevents the service provider from properly summarizing customer routes if it is also using private address space.

- Private IP addressing prevents service provider applications such as MPLS from operating properly in an Internet-supporting environment.

1. In service provider network It is recommended to to redistribute BGP routes into IGP protocol. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☒ true

☐ false

1. When using next hop self feature on the PE router in the service provider network, IGP can be relieved of the burden of carrying information about the access link. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ true

☐ false

1. Which three requirements are key to properly scaling BGP in a service provider environment? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

- ☐ IBGP full-mesh scaling tools to reduce duplicate traffic within the AS
- ☐ summarization of customer routes to reduce the number of prefixes that are carried
- ☐ improvement in BGP convergence time by using the IGP for route propagation within the provider AS
- ☐ proper scaling of the AS-wide routing policy to ease administration and maintenance requirements

1. Which three requirements are key to properly scaling BGP in a service provider environment? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

- ☐ IBGP full-mesh scaling tools to reduce duplicate traffic within the AS
- ☐ summarization of customer routes to reduce the number of prefixes that are carried
- ☐ improvement in BGP convergence time by using the IGP for route propagation within the provider AS
- ☐ proper scaling of the AS-wide routing policy to ease administration and maintenance requirements

1. In the traditional BGP approach, ensuring consistent routing information was achieved by establishing a full mesh of IBGP sessions between all routers within the AS. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ true

☐ false

1. What is the main problem that is solved by implementing BGP route reflectors? (Source: "Introducing and Designing Route Reflectors")

- the large number of routes that are carried in the IGP when BGP is deployed
- the ability of BGP to scale a single AS in a large network
- the need for a homogeneous method of applying policies to routes that are carried through an AS
- the ability to support service-level parameters with greater ease

1. How does a route reflector modify the IBGP split-horizon rule?  
(Source: "Introducing and Designing Route Reflectors")

- forwards EGP updates to all peers (IBGP and EGP)
- treats all neighbors as EGP peers, which eliminates the IBGP mesh requirements
- forwards IBGP updates from clients to other IBGP neighbors
- appends the cluster-ID to the AS path, which allows peers to be treated as EGP neighbors



1. When incoming BGP update is received from EBGP peer on route reflector, the update is forwarded to both IBGP and EBGP peers. True or false? (Source: "Introducing and Designing Route Reflectors")

☐ true

☐ false

1. Why are redundant route reflectors mandatory in any high-availability network design? (Source: "Introducing and Designing Route Reflectors")

- All neighbors peer with the route reflector, and a large number of neighbors can make the route reflector router unstable.

- EBGp peers can inject BGP updates into the AS only through the route reflector.

- Route reflectors maintain more routing information, which makes them more prone to congestion and failure.

- Clients can form IBGP relationships only with the route reflector.

1. What is the main reason for implementing redundant route reflectors with clusters? (Source: "Introducing and Designing Route Reflectors")

- to eliminate routing loops in redundant configurations

- to limit the number of neighbor sessions with each route reflector

- to provide another scalability mechanism targeted at removing the IBGP full-mesh requirement

- to enhance security within the AS

1. How does the originator-ID attribute assist in the elimination of routing loops that are caused by redundant route reflector designs? (Source: "Introducing and Designing Route Reflectors")

- If the originator-ID matches the router-ID of the reflector, local preference is set on the route to make it a backup.

- The originator-ID attribute is set to the cluster-ID to ensure that a route traverses the AS only one time.

- A router that receives a route in which the originator-ID matches its router-ID will ignore that route.

- The originator-ID allows the router to know if the route originated locally or from an external source so that administrative distance rules for the route can be verified.

1. What can occur if a client has IBGP neighbor relationships with other routers that are not configured as route reflectors? (Source: "Introducing and Designing Route Reflectors")

- This is an invalid configuration.

- The client will notify the route reflector and be promoted to a route reflector as well.

- Routing black holes can occur and cause lost traffic inside the AS.

- Unnecessary routing traffic will be generated.

1. Which potential problem can occur if a client does not have an IBGP session with all route reflectors in a cluster? (Source: "Introducing and Designing Route Reflectors")

☐ This is an invalid configuration.

☐ The client might not receive all BGP routes.

☐ EBGP routes that are received by the client will not be distributed properly throughout the AS.

☐ Duplicate routing traffic will be sent to the client.

1. Which problem are hierarchical route reflectors designed to solve?  
(Source: "Introducing and Designing Route Reflectors")

- lack of a consistent application of security and routing policies throughout the AS

- scalability of autonomous systems in very large routing domains

- routing loops caused by redundant cluster configurations

- administrative overhead when you are implementing router reflector network designs

1. Which two BGP parameters do you have to configure on a route reflector? (Choose two.) (Source: "Configuring and Monitoring Route Reflectors")

- ☐ cluster-ID
- ☐ originator-ID
- ☐ cluster-list
- ☐ route reflector clients



1. What are three migration steps that are required to convert from a fully meshed IBGP AS to an AS that is based on route reflectors? (Choose three.) (Source: "Configuring and Monitoring Route Reflectors")

- ☐ remove unnecessary IBGP sessions
- ☐ configure the clients on the route reflectors
- ☐ configure IBGP sessions between route reflector clients
- ☐ configure the cluster-ID on the route reflectors

1. Which command should you use to identify route reflector clients without inspecting the router configuration? (Source: "Configuring and Monitoring Route Reflectors")

☐ **show ip bgp *prefix***

☐ **show ip bgp neighbors**

☐ **show ip bgp clients**

☐ **show ip bgp summary**

1. The command **neighbor *ip-address* route-reflector-client** command should be used on client. True or false? (Source: "Configuring and Monitoring Route Reflectors")

☒ true

☐ false

## Answer Key

1. Which three characteristics are common to typical service provider networks? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ The provider network uses two IGPs, one for customer routes and one for internal provider routes.

☒ Service providers exchange routes with other providers using BGP.

☒ Service providers run IBGP within their network in addition to their IGP requirements.

☒ Service providers typically use either static routes or EBGp with their customers.

1. What is the typical role of an IGP within a service provider network?  
(Source: "Scaling IGP and BGP in Service Provider Networks")

The IGP carries customer routes for redistribution into BGP at the provider edge.

The IGP advertises a default route to customers of the service provider.

The IGP resolves next-hop IP addresses.

The IGP carries BGP routes across the provider network.

1. The IGP protocol is not needed in the service provider network, as information about next hop attribute gets advertised via BGP updates. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ true

☐ false

1. Why should you avoid the use of private IP addressing in service provider networks? (Source: "Scaling IGP and BGP in Service Provider Networks")

Private addressing can prevent customer network troubleshooting utilities such as traceroute from functioning correctly.

Private IP addressing is not allowed on the Internet and will not function in a service provider network.

Private IP addressing prevents the service provider from properly summarizing customer routes if it is also using private address space.

Private IP addressing prevents service provider applications such as MPLS from operating properly in an Internet-supporting environment.

1. In service provider network It is recommended to to redistribute BGP routes into IGP protocol. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☒ true

☐ false



1. When using next hop self feature on the PE router in the service provider network, IGP can be relieved of the burden of carrying information about the access link. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

true

false

1. Which three requirements are key to properly scaling BGP in a service provider environment? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

- ☒ IBGP full-mesh scaling tools to reduce duplicate traffic within the AS
- ☒ summarization of customer routes to reduce the number of prefixes that are carried
- ☐ improvement in BGP convergence time by using the IGP for route propagation within the provider AS
- ☒ proper scaling of the AS-wide routing policy to ease administration and maintenance requirements

1. Which three requirements are key to properly scaling BGP in a service provider environment? (Choose three.) (Source: "Scaling IGP and BGP in Service Provider Networks")

- ☒ IBGP full-mesh scaling tools to reduce duplicate traffic within the AS
- ☒ summarization of customer routes to reduce the number of prefixes that are carried
- ☐ improvement in BGP convergence time by using the IGP for route propagation within the provider AS
- ☒ proper scaling of the AS-wide routing policy to ease administration and maintenance requirements

1. In the traditional BGP approach, ensuring consistent routing information was achieved by establishing a full mesh of IBGP sessions between all routers within the AS. True or false? (Source: "Scaling IGP and BGP in Service Provider Networks")

☐ true

☐ false

1. What is the main problem that is solved by implementing BGP route reflectors? (Source: "Introducing and Designing Route Reflectors")

- the large number of routes that are carried in the IGP when BGP is deployed
- the ability of BGP to scale a single AS in a large network
- the need for a homogeneous method of applying policies to routes that are carried through an AS
- the ability to support service-level parameters with greater ease

1. How does a route reflector modify the IBGP split-horizon rule?  
(Source: "Introducing and Designing Route Reflectors")

- forwards EGP updates to all peers (IBGP and EGP)

- treats all neighbors as EGP peers, which eliminates the IBGP mesh requirements

- forwards IBGP updates from clients to other IBGP neighbors

- appends the cluster-ID to the AS path, which allows peers to be treated as EGP neighbors

1. When incoming BGP update is received from EBGP peer on route reflector, the update is forwarded to both IBGP and EBGP peers. True or false? (Source: "Introducing and Designing Route Reflectors")

☐ true

☐ false

1. Why are redundant route reflectors mandatory in any high-availability network design? (Source: "Introducing and Designing Route Reflectors")

All neighbors peer with the route reflector, and a large number of neighbors can make the route reflector router unstable.

EBGP peers can inject BGP updates into the AS only through the route reflector.

Route reflectors maintain more routing information, which makes them more prone to congestion and failure.

Clients can form IBGP relationships only with the route reflector.



1. What is the main reason for implementing redundant route reflectors with clusters? (Source: "Introducing and Designing Route Reflectors")

- to eliminate routing loops in redundant configurations

- to limit the number of neighbor sessions with each route reflector

- to provide another scalability mechanism targeted at removing the IBGP full-mesh requirement

- to enhance security within the AS

1. How does the originator-ID attribute assist in the elimination of routing loops that are caused by redundant route reflector designs? (Source: "Introducing and Designing Route Reflectors")

If the originator-ID matches the router-ID of the reflector, local preference is set on the route to make it a backup.

The originator-ID attribute is set to the cluster-ID to ensure that a route traverses the AS only one time.

A router that receives a route in which the originator-ID matches its router-ID will ignore that route.

The originator-ID allows the router to know if the route originated locally or from an external source so that administrative distance rules for the route can be verified.

1. What can occur if a client has IBGP neighbor relationships with other routers that are not configured as route reflectors? (Source: "Introducing and Designing Route Reflectors")

• This is an invalid configuration.

• The client will notify the route reflector and be promoted to a route reflector as well.

• Routing black holes can occur and cause lost traffic inside the AS.

• Unnecessary routing traffic will be generated.

1. Which potential problem can occur if a client does not have an IBGP session with all route reflectors in a cluster? (Source: "Introducing and Designing Route Reflectors")

☐ This is an invalid configuration.

☐ The client might not receive all BGP routes.

☐ EBGP routes that are received by the client will not be distributed properly throughout the AS.

☐ Duplicate routing traffic will be sent to the client.

1. Which problem are hierarchical route reflectors designed to solve?  
(Source: "Introducing and Designing Route Reflectors")

- lack of a consistent application of security and routing policies throughout the AS

- scalability of autonomous systems in very large routing domains

- routing loops caused by redundant cluster configurations

- administrative overhead when you are implementing router reflector network designs

1. Which two BGP parameters do you have to configure on a route reflector? (Choose two.) (Source: "Configuring and Monitoring Route Reflectors")

☒ cluster-ID

☐ originator-ID

☐ cluster-list

☒ route reflector clients

1. What are three migration steps that are required to convert from a fully meshed IBGP AS to an AS that is based on route reflectors? (Choose three.) (Source: "Configuring and Monitoring Route Reflectors")

- ☒ remove unnecessary IBGP sessions
- ☒ configure the clients on the route reflectors
- ☐ configure IBGP sessions between route reflector clients
- ☒ configure the cluster-ID on the route reflectors

1. Which command should you use to identify route reflector clients without inspecting the router configuration? (Source: "Configuring and Monitoring Route Reflectors")

☐ **show ip bgp *prefix***

☐ **show ip bgp neighbors**

☐ **show ip bgp clients**

☐ **show ip bgp summary**



1. The command **neighbor *ip-address* route-reflector-client** command should be used on client. True or false? (Source: "Configuring and Monitoring Route Reflectors")

☒ true

☐ false

# Optimizing BGP Scalability

## Introduction

[BGP](#) is designed for reliability and scalability. As such, it has become the standard protocol that is used to carry the more than 600,000 prefixes in the Internet today. BGP also has a tremendous amount of flexibility with regard to administrative policy controls, route selection, performance tuning, and scalability features.

You will learn about advanced BGP configuration tools that are designed to improve BGP scalability and performance. Tools that are discussed in this module include convergence time reduction features, limiting the number of prefixes, peer groups, and route dampening.

Upon completing this module, you will be able to:

- Configure Cisco IOS performance improvements to reduce BGP convergence time
- Configure BGP to limit the number of prefixes that are received from a neighbor
- Use BGP peer groups to share common configuration parameters between multiple BGP peers
- Use route dampening to minimize the impact of unstable routes

# Improving BGP Convergence

## Overview

As the number of routes in the Internet increases, demands on router [CPU](#) and memory resources on the router in a service provider will increase. [BGP](#) processing affects both router resources and network convergence time. It is important that the network convergence is as fast as possible to ensure accurate routing information between domains. It is also important that router resources are optimized whenever possible. Cisco IOS performance improvements for BGP are designed to aid network administrators in achieving these goals.

In this lesson, you will learn about various Cisco IOS performance improvements that have been designed to reduce BGP convergence time. Included in this lesson are discussions of convergence, BGP routing processes, and the effects of BGP routing processes on router CPU resources. The lesson also discusses the commands that are required to configure and monitor BGP for various Cisco IOS performance improvements. Improvements that are mentioned are [PMTU](#) discovery, input hold queue, BGP [PIC](#), [BFD](#), BGP [NSF](#) awareness, BGP scan time, advertisement interval, keepalive, and hold-down timers.

Upon completing this lesson, you will be able to:

- Describe convergence in BGP networks
- Describe the BGP router processes and their functions
- Describe the effects of BGP processes on router CPU resources
- Describe the features that can be used to improve BGP convergence
- Describe using PMTU discovery
- Describe increasing the input queue depth
- Describe the BGP prefix independent convergence feature and configuration
- Describe the BGP BFD feature and configuration
- Describe the function of the BGP Nonstop Forwarding Awareness feature
- Explain BGP scan timer configuration
- Explain BGP advertisement interval configuration
- Explain BGP keepalive and hold-down timer configuration

## BGP Convergence

As the number of routes in the Internet routing table grows, service providers and large enterprise customers are experiencing a dramatic increase in the time that [BGP](#) takes to converge. Networks that once converged in 10 or 15 minutes may now take up to an hour in some cases, and even longer in extreme situations. In general, convergence is defined as the process of bringing all routing tables to a state of consistency.

- As the number of routes in the Internet routing table grows, the time it takes for BGP to converge increases.
- The Internet currently contains more than 300,000 prefixes.
- Network convergence times can range from 10 minutes to more than 1 hour.
- BGP is considered converged when:
  - All routes have been accepted.
  - All routes have been installed in the routing table.
  - The input queue and output queue for all peers is 0.
  - The table version for all peers equals the table version of the BGP table.

The BGP routing protocol is considered converged when the following conditions are true:

- All routes have been accepted.
- All routes have been installed in the routing table.
- The input queue and output queue for all peers is 0.
- The table version for all peers equals the table version of the BGP table.

Convergence time is an important consideration in a network, because nonconverged networks can cause routing loops, packet delays, and even packet loss as a result of black holes.

## BGP Processes

In general, a Cisco IOS process consists of the individual threads and associated data that perform router tasks, such as system maintenance, packet switching, and implementing routing protocols.

**BGP** consists of several processes, each of them running at different times, depending on the task that is handles.

| Process     | Description                                                                                                                                                                                               | Interval                                                                     |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| BGP open    | Performs BGP peer establishment.                                                                                                                                                                          | At initialization, when establishing a TCP connection with a BGP peer.       |
| BGP I/O     | Handles queuing and processing of BGP packets (updates and keepalives).                                                                                                                                   | As BGP control packets are received.                                         |
| BGP scanner | Walks the BGP table and confirms reachability of the next hops. BGP scanner also checks conditional advertisement to determine whether BGP should advertise condition prefixes. Performs route dampening. | Every 60 seconds.                                                            |
| BGP router  | Calculates the best BGP path and processes any route changes. It also sends and receives routes, establishes peers, and interacts with the routing information base.                                      | Once per second and when adding, removing, or soft-reconfiguring a BGP peer. |

- BGP scanner and BGP router are responsible for many calculations and can lead to high CPU utilization.

Several processes that are executed on the router enable BGP to run. You can use the **show process cpu | include BGP** command to see the volume of [CPU](#) resources that are consumed (utilization) because of running BGP processes.



A thread is an information placeholder that allows a single process to be halted (interrupted) on the router so that the CPU can service another process. The information that is contained within the thread allows the interrupted process to restart exactly where it left off when the CPU is ready to continue to service that process thread.

The figure lists the function of each of the BGP router processes and how often each process is executed on the router. It shows that each process runs at different times, depending on the tasks that this specific process handles. Because BGP scanner and BGP router are responsible for many calculations, you may notice high CPU utilization during the running of either one of these processes.

## CPU Effects of BGP Processes

Running [BGP](#) router processes affects router [CPU](#) resources.

BGP scanner process:

- High CPU utilization stemming from the BGP scanner process can be expected for short durations on a router carrying a large Internet routing table.
- While the BGP scanner runs, low-priority processes need to wait a longer time to access the CPU.

BGP router process:

- The BGP router process runs about once per second to check for work.
- The BGP router consumes all free CPU cycles.

On routers that carry a large Internet routing table, you can expect high CPU utilization stemming from the BGP scanner process for short periods of time. Once per minute, the BGP scanner "walks" (scans) the BGP routing table and performs important maintenance tasks. These tasks include checking the next hop that is referenced in the BGP table of the router and verifying that the next-hop devices can be reached. Thus, a large BGP table takes an equally large amount of time to be walked and validated.

The BGP scanner walks the BGP routing table to update any data structures and walks the table for route redistribution purposes. In this context, the routing table is also known as the [RIB](#), which the router outputs when the **show ip route** command is executed. Both tables are stored separately in the router memory and can be very large, thus consuming CPU and memory resources.

The BGP scanner runs through the entire BGP table. So the duration of the high CPU utilization condition that the BGP scanner process causes varies with the number of neighbors and the number of routes that are learned per neighbor.

While the BGP scanner runs, low-priority processes need to wait a longer time to access the CPU. One low-priority process controls [ICMP](#) packets such as pings. Packets that are destined to or have originated from the router may experience higher than expected latency because the ICMP process must wait behind the BGP scanner. The BGP scanner process runs for some time, and is suspended, then ICMP runs and is suspended, then the BGP scanner runs, and so on. In contrast, pings sent through a router should be switched via Cisco Express Forwarding (CEF) and should not experience any additional latency. When you are troubleshooting periodic spikes in latency, compare forwarding times for packets that are forwarded through a router versus packets that are processed directly by the CPU on the router.

The BGP router process runs about once per second to check for work. BGP convergence defines the duration between the time when the first BGP peer is established and the point at which BGP is converged. To ensure the shortest possible convergence times, the BGP router consumes all free CPU cycles. However, after it starts, it relinquishes (or suspends) the CPU intermittently.

## CPU Effects of BGP Processes Example

Convergence time is a direct measurement of how long the BGP router process runs on the CPU, not the total time that the process is actually running. This example investigates the high CPU utilization condition on a router during BGP convergence as BGP exchanges prefixes with two [EBGP](#) peers.

- Capture a baseline for normal CPU utilization before starting the test.

```
R1# show process cpu
CPU utilization for five seconds: 0%/0%; one minute: 4%; five minutes: 5%
```

- After the test starts, the CPU reaches 100 percent utilization. The **show process cpu** command shows that the BGP router causes high CPU condition, denoted by 139 (the Cisco IOS process ID for the BGP router) in the following output:

```
R1# show process cpu
CPU utilization for five seconds: 100%/0%; one minute: 99%; five minutes:
81%
[output omitted] 139 6795740 1020252 6660 88.34% 91.63% 74.01% 0
BGP Router
```

- Monitor the router by capturing multiple outputs of the **show ip bgp summary** and **show process cpu** commands during the event. The **show ip bgp summary** command captures the state of the BGP neighbors.

```
R1# show ip bgp summary
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
192.168.12.2 4 64512 309453 157389 19981 0 253 22:06:44 11163
3
172.16.11.11 4 65101 188934 1047 40081 41 0 00:07:51 58430
```

- When the router completes prefix exchange with its BGP peers, the CPU utilization rates should return to normal levels. The computed 1-minute and 5-minute averages will settle back down as well but may show higher than normal levels for a longer period than the 5-second rate.

```
R1# show process cpu
CPU utilization for five seconds: 3%/0%; one minute: 82%; five minutes: 91%
%
```

- Using the output from the **show** commands will allow you to compute the BGP convergence time. In particular, the Up/Down column of the **show ip bgp summary** command is compared to the start and stop times of the high CPU utilization condition. Typically, BGP convergence can take several minutes when routers exchange a large Internet routing table.

## Improving BGP Convergence

**BGP** convergence can often be an issue in networks requiring quick propagation of routing information.

- Queuing to TCP peer connections
  - BGP automatically queues data aggressively from the BGP output queue
- Deploy BGP peer groups
  - Simplifies BGP configuration and enhances BGP scalability.
- Enable the path MTU feature
  - Improves efficiency by dynamically determining the largest MTU that you can use without creating packets that need to be fragmented.
- Implement BFD:
  - Reduces BGP convergence by fast detection of neighbor failure.
- Implement BGP PIC:
  - Reduces convergence by storing BGP backup/alternate path in RIB and FIB.
- Increase interface input queues
  - Improves convergence by reducing dropped TCP ACKs.
- Enable BGP NSF Awareness feature
  - Forwards packets even during failure and thus, saves resources.

To reduce BGP convergence time and the high **CPU** utilization that a running BGP process causes, the following performance improvement features are available:

- **Queuing to TCP peer connections:** Instead of queuing data once per second, BGP now queues data aggressively from the BGP output queue to the **TCP** socket for each peer until the output queues have drained completely. Because BGP now sends at a faster rate, it converges more quickly.
- **BGP peer groups:** The major benefit of specifying a BGP peer group is that it reduces the volume of system resources (CPU and memory) that are used in BGP update generation. Peer groups also simplify BGP configuration. Many repetitive configuration elements (such as filters) are applied by the router only once (to the peer group) instead of applying them to each neighbor. Peer groups allow the routing table to be checked only once and allow updates to be replicated to all other in-sync peer group members. It depends on the number of peer group members, the number of prefixes in the table, and the number of prefixes that are advertised. Consequently they can significantly reduce router resource requirements.
- **Bidirectional Forwarding Detection:** **BFD** is a detection protocol designed to provide fast forwarding path failure detection times for all media types, encapsulations, topologies, and routing protocols.
- **BGP Prefix Independent Convergence:** The BGP **PIC** feature improves BGP convergence after a network failure. This convergence is applicable to both core and edge failures and can be used in both IP and **MPLS** networks. The BGP PIC feature creates and stores a backup/alternate path in the **RIB**, **FIB**, and Cisco Express Forwarding (CEF). When a failure is detected, the backup/alternate path can immediately take over, thus enabling fast failover.
- **Path MTU feature:** All TCP sessions are bounded by a limit on the number of bytes that a single packet can transport. This limit, which is known as the **MSS**, is 536 bytes by default. In other words, TCP breaks up packets in a transmit queue into 536-byte chunks before passing packets down to the IP layer. The advantage of a 536-byte MSS is that packets are not likely to be fragmented at an IP device along the path to the destination, because most links use an **MTU** of at least 1500 bytes. The disadvantage is that smaller packets increase the amount of bandwidth that is used for transport overhead. Because BGP builds a TCP connection to all peers, a 536-byte MSS affects BGP convergence times. The solution is to enable the **PMTU** feature. You can use this feature to dynamically determine how large the MSS value can be without creating packets that need to be fragmented. PMTU allows TCP to determine the smallest MTU size among all links in a TCP session. TCP then uses this MTU value, minus room for the IP and TCP headers, as the MSS for the session.
- **Increase interface input queues:** If BGP is advertising thousands of routes to many neighbors, TCP must transmit thousands of packets. BGP peers receive these packets and send TCP **ACKs** to the advertising BGP speaker, causing the BGP speaker to receive a flood of TCP ACKs in a short time. If the ACKs arrive



at a rate that is too high for the router CPU, packets back up in inbound interface queues. By default, router interfaces use an input queue size of 75 packets. In addition, special control packets such as BGP updates use a special queue with [SPD](#). This special queue holds 100 packets. During BGP convergence, TCP ACKs can quickly fill the 175 spots of input buffering, causing newly arriving packets to be dropped. On routers with 15 or more BGP peers that also exchange the full Internet routing table, more than 10,000 drops per interface per minute may be seen. Increasing the interface input queue depth helps reduce the number of dropped TCP ACKs, reducing the amount of work that BGP must do to converge.

- **BGP Nonstop Forwarding Awareness feature:** [NSF](#) awareness allows a NSF-aware router to assist NSF-capable and NSF-aware neighbors to continue forwarding packets during a switchover operation or during a well-known failure condition. The router-hold timer sets the maximum time that the NSF-aware router will hold known routes for an NSF-capable neighbor. The deployment of BGP NSF awareness can improve the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

BGP convergence can also be improved to some extent by:

- Lowering scan time interval for the BGP scanner process.
- Lowering advertisement interval between BGP neighbors.
- Lowering keepalive and hold-down timers.

Limitation:

- Not recommended in routers dealing with large BGP tables.
- Could lead to CPU or memory exhaustion.
- Lower hold-down timers could lead to undesired session terminations.

You also need to improve BGP convergence in certain scenarios; for example, in networks using the conditional advertisement feature. There are four extra BGP parameters that you can use to influence BGP convergence speed:

- **Scan time:** Controlling the BGP scanner process, responsible for verifying information in the BGP table
- **Advertisement interval:** Controlling the rate at which successive advertisements are sent to a BGP neighbor
- **Keepalive timers:** Controlling the BGP session by sending keepalive messages at specific time interval.
- **Hold-down timers:** Controlling the BGP session by waiting for successive keepalive message from a BGP neighbor.

Network administrators must take care when configuring these parameters. Setting the values too low for a specific network environment could lead to a significant consumption of router resources. The larger the BGP tables and the more unstable the BGP network, the greater the danger of exhausting the resources of a router. Lowering the keepalive and hold-down timers could lead to undesired session terminations when keepalives are not received due to network congestion.

## PMTU Discovery

PMTU discovery feature can be used to reduce BGP convergence. PMTU discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection.

PMTU discovery is used to automatically determine TCP MSS used for TCP connections from a router.

- Prior to Cisco IOS Release 15.0, the default TCP MSS value for BGP was 536 bytes.
- From Cisco IOS Release 15.0, PMTU is enabled by default.
- Small TCP MSS affects BGP convergence:
  - Higher TCP MSS can improve BGP convergence.

```
router(config)# ip tcp path-mtu-discovery [age-timer {minutes | infinite}]
```

- The command enables the PMTU discovery feature for all new TCP connections from the router.
  - The age timer is a time interval for how often TCP re-estimates the path MTU with a larger MSS (default is 10 minutes).

PMTU discovery works by setting the [DF](#) option bit in the IP headers of outgoing packets. Then, any device along the path whose [MTU](#) is smaller than the packet will drop it. Then it will send back an [ICMP](#) Fragmentation Needed (Type 3, Code 4) message containing its MTU, allowing the source host to reduce its PMTU appropriately. The process repeats until the MTU is small enough to traverse the entire path without fragmentation.

The default [TCP MSS](#) for BGP traffic is 536 bytes. Enabling PMTU discovery, and thus using a higher MSS for BGP traffic, can significantly improve BGP convergence, since it takes fewer packets to send BGP updates.

To enable the PMTU discovery feature for all new TCP connections from the router, use the **ip tcp path-mtu-discovery** global configuration command. To disable the function, use the **no** form of this command:

```
ip tcp path-mtu discovery [age-timer {minutes | infinite}]
```

### Syntax Description

| Parameter                        | Description                                                                                                                                                   |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>age timer</b> <i>minutes</i>  | (Optional) Time interval (in minutes) after which TCP re-estimates the PMTU with a larger MSS. The maximum interval is 30 minutes; the default is 10 minutes. |
| <b>age-timer</b> <i>infinite</i> | (Optional) Turns off the age timer.                                                                                                                           |

The age timer is a time interval for how often TCP re-estimates the PMTU with a larger MSS. The default value of the age timer is 10 minutes, but it can be manually configured up to 30 minutes or disabled (set to infinite). If the MSS that is used for the connection is smaller than the peer connection can handle, the router will attempt to use a larger MSS each time that the age timer expires. The discovery process is stopped when either the sent MSS is as large as the peer negotiated or the user has disabled the timer on the router. You can turn off the age timer by setting it to "infinite."

## Monitoring PMTU Discovery

To verify the TCP MSS that is used between BGP neighbors, use the **show ip bgp neighbors**. You can use **include max data** filter to display relevant output only, as shown in the example.

```
router# show ip bgp neighbor | include max data
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
Datagrams (max data segment is 536 bytes):
```

- The default MSS is 536 bytes (before Cisco IOS Release 15.0).

```
router# show ip bgp neighbor | include max data
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
Datagrams (max data segment is 1460 bytes):
```

- After enabling of the PMTU discovery feature, the MSS has been increased.

The first example shows the default size of the MSS, 536 bytes, before the PMTU discovery feature is enabled on the router. After using the **ip tcp path-mtu-discovery** command to enable PMTU discovery, the router dynamically determines how large the MSS can be without creating IP packets that require fragmentation. In the example at the bottom, the output shows that the PMTU feature has been enabled. The **show ip bgp neighbors | include max data** command has been used to determine that the PMTU discovery feature has set the MSS to 1460 bytes.

## Increasing Input Queue Depth

Another feature can be used to improve [BGP](#) convergence is to increase input queue depth. BGP routers might experience packet drops on an interface due to large number of [TCP ACK](#) segments, which are used to acknowledge receipt of BGP updates.

- Input queue on an interface specifies how many packets can be queued before dropping the packets.
- BGP routers with several peers might experience packet drops on an interface due to many TCP ACK segments.

```
router(config-if)# hold-queue length in
```

- This command limits the size of the IP queue on an interface.
  - The default input hold-queue limit is 75 packets, configurable from 0 to 65,535 packets.
  - A length of 1000 will normally resolve problems that are caused by input queue drops of TCP ACKs.

Each interface owns an input queue into which incoming packets are placed to await processing by the router. Frequently, the rate at which incoming packets are placed in the input queue exceeds the rate at which the router can process the packets. Each input queue has a size that indicates the maximum number of packets that can be placed in the queue. After the input queue becomes full, the interface drops any new incoming packets.

To specify the size of the IP input or output queue on an interface, use the **hold-queue** command in interface configuration mode. To restore the default values for an interface, use the **no** form of this command with the appropriate keyword.

```
hold-queue length {in | out}
```

### Syntax Description

| Parameter     | Description                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>length</i> | Integer that specifies the maximum number of packets in the queue. The range of allowed values is 0 to 65535.                                                                                                |
| <b>in</b>     | Specifies the input queue. The default is 75 packets. For asynchronous interfaces, the default is 10 packets. These limits prevent a malfunctioning interface from consuming an excessive amount of memory.  |
| <b>out</b>    | Specifies the output queue. The default is 40 packets. For asynchronous interfaces, the default is 10 packets. These limits prevent a malfunctioning interface from consuming an excessive amount of memory. |



Several considerations should be taken into account when increasing interface input queues. Increasing the interface input queues will increase memory utilization.



Increasing the hold queue can have detrimental effects on network routing and response times. For protocols that use SEQ or ACK packets to determine round-trip times, do not increase the output queue. Dropping packets instead informs hosts to slow down transmissions to match available bandwidth. This approach is generally better than having duplicate copies of the same packet within the network (which can happen with large hold queues).



The Cisco 12000 Series now uses a default [SPD](#) headroom value of 1000. It retains the default input queue size of 75. Use the **show spd** command to view these special input queues.

## Monitoring PMTU Input Queue Depth

Use the **show interface {interface-identifier}** command to displays the current input

queue levels and the number of incoming packets dropped.

```
router# show interface GigabitEthernet0/0
GigabitEthernet0/0 is up, line protocol is up
 Hardware is AmdP2, address is aabb.cc00.4900 (bia aabb.cc00.4900)
 MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
 reliability 255/255, txload 1/255, rxload 1/255
 Encapsulation ARPA, loopback not set
 Keepalive set (10 sec)
 ARP type: ARPA, ARP Timeout 04:00:00
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
 Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
```

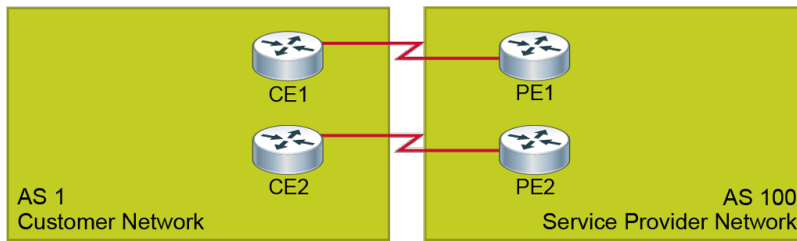
- Displays interface information, including input queue depth.

The input queue w/x/y/z counter displays the current input queue. w is the current number of packets in the input queue and x is the queue depth. The drops counter, y, indicates the number of incoming packets that have been dropped. In the example, there is no packet in the input queue, the depth of queue is left on default, 75, and no packet were dropped.

If the current number of packets in the input queue is consistently at, or greater than, 80 percent of the current size of the input queue, the size of the input queue may require tuning to accommodate the rate of incoming packets. Even if the current number of packets in the input queue never seems to approach the size of the input queue, bursts of packets may still be overflowing the queue. If the drops counter is increasing at a high rate, the size of the input queue may require tuning to accommodate the bursts.

## BGP Prefix Independent Convergence

**BGP** Prefix Independent Convergence feature improves BGP convergence after a network failure. It calculates a second best path, along with the primary best path and stores both paths in the Cisco Express Forwarding (CEF).



BGP prefix independent convergence characteristics:

- PIC enhances BGP convergence, regardless of the number of BGP prefixes.
- PIC stores the BGP backup/alternate path for each prefix in BGP, RIB, and FIB tables.
- When the primary goes down, Cisco Express Forwarding quickly selects a different egress port for the affected destination.

Under normal circumstances, BGP can take several seconds to a few minutes to converge after a network change. At a high level, BGP goes through the following process:

1. BGP learns of failures through either **IGP** or **BFD** events or interface events.
2. BGP withdraws the routes from the **RIB**, and the RIB withdraws the routes from the **FIB** and distributed FIB. This process clears the data path for the affected prefixes.
3. BGP sends withdraw messages to its neighbors.
4. BGP calculates the next best path to the affected prefixes.
5. BGP inserts the next best path for affected prefixes into the RIB, and the RIB installs them in the FIB and distributed FIB.

This process takes a few seconds or a few minutes to complete. The time depends on the latency of the network, the convergence time across the network, and the local load on the devices. The data plane converges only after the control plane converges.

The BGP **PIC** functionality is achieved by additional functionality in the BGP, RIB, and CEF.

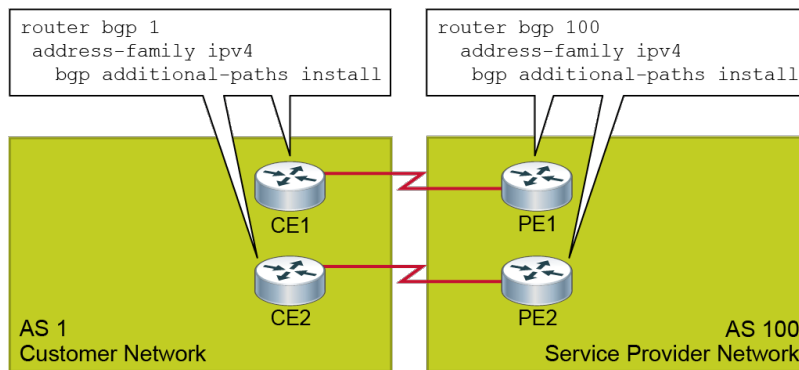
BGP PIC affects prefixes under **IPv4** and **VPNv4** address families. For those prefixes, BGP calculates a second best path, along with the primary best path. (The second best path is called the backup/alternate path.) BGP installs the best path and the backup/alternate paths for the affected prefixes into the BGP RIB. The backup/alternate path provides a fast reroute mechanism to counter a single network failure.

For BGP PIC, RIB installs an alternate path per route if one is available. With the BGP PIC functionality, a RIB that selects a BGP route containing a backup/alternate path installs the backup/alternate path with the best path.

With BGP PIC, CEF stores an alternate path per prefix. When the primary path goes down, CEF searches for the backup/alternate path in a prefix-independent manner. CEF also listens to BFD events to rapidly detect local failures. Upon detection of a failure, CEF detects the alternate next hop for all prefixes that are affected by the failure.

## BGP PIC Configuration

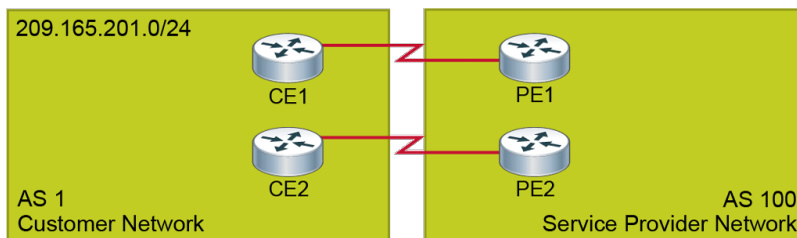
To enable BGP PIC on Cisco IOS Software routers, first enter BGP configuration mode using the **router bgp** command. Then, enter the appropriate address family configuration mode and use the **bgp additional-paths install** command to enable BGP PIC.



In the example, CE1 and CE2 are configured with the BGP PIC feature. BGP computes PE1 as the best path and PE2 as the backup/alternate path, and installs both the routes into the RIB and CEF plane.

No policies should be set on CE1 and CE2 for the EBGPs. Both CE routers must point to the EBGPs. On CE1, the next hop to reach the Internet is through PE1. On CE2, the best path to reach the Internet is PE2. CE2 advertises itself as the next hop to CE1, and CE1 does the same to CE2. As a result, CE1 has two paths for the specific prefix. It usually selects the directly connected EBGPs path over the IBGP path, according to the best path selection rules. Similarly, CE2 has two paths, an EBGPs path through PE2 and an IBGP path through CE1-PE1.

If the CE1-PE1 link or PE1 goes down, BGP recomputes the best path, removing the next hop PE1 from RIB and reinstalling CE2 as the next hop into the RIB and CEF. CE1 automatically gets a backup/alternate repair path into CEF, and the traffic loss during forwarding is now in subseconds, achieving fast convergence.



```
PE1# show ip bgp
BGP table version is 65, local router ID is 10.0.3.81
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 209.165.201.0/24 1.0.0.101 0 100 0 1 i
*b i 10.0.100.2 0 100 0 1 i
```

You can verify the backup route by issuing **show ip bgp** command. In the example, you can see that PE1 router has two paths to reach the 209.165.201.0/24 network at the customer side. The internal route, via PE2 router, is marked as the backup route.

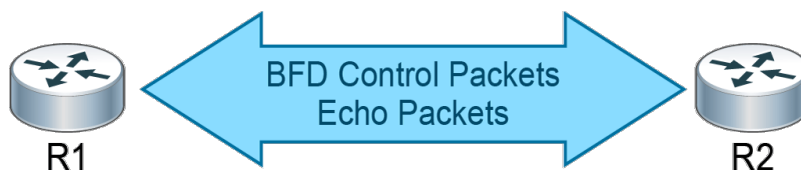
Additionally, you can verify that the backup path is really stored in the CEF, by issuing **show ip cef prefix detail** command. The backup route is labeled as "repair." Without BGP PIC feature enabled, only one path is stored in the CEF.

```
PE1# show ip cef 209.165.201.0/24 detail
209.165.201.0/24, epoch 0, flags rib only nlabel, rib defined all labels
recursive via 1.0.0.101
 attached to Serial0
recursive via 10.0.100.2, repair
 attached to Ethernet0/0
```

## Bidirectional Forwarding Detection for BGP

[BFD](#) provides a low-overhead, short-duration method of detecting failures in the forwarding path between two adjacent routers, including the interfaces, data links, and forwarding planes. BFD is a detection protocol that you enable at the interface and routing protocol levels. Cisco supports the BFD asynchronous mode, which depends on the sending of BFD control packets between two systems to activate and maintain BFD neighbor sessions between routers. Therefore, in order for a BFD session to be created, you must configure BFD on both BFD peers. Once BFD has been enabled on the interfaces and at the router level for the appropriate routing protocols, a BFD session is created, BFD timers are negotiated, and the BFD peers will begin to send BFD control packets to each other at the negotiated interval.

- Extremely lightweight hello protocol that uses UDP to test bidirectional communication.
- Used to detect failures in the forwarding path between two adjacent routers.
- Millisecond resolution of forwarding plane failure.
- Relies on routing protocols to detect neighbors.

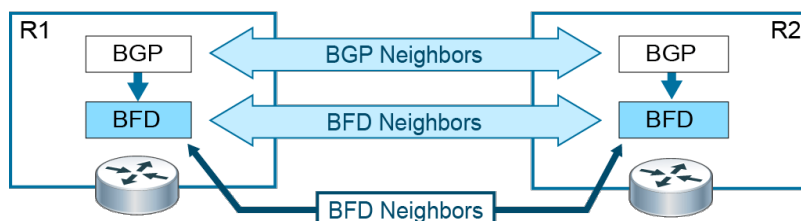


BFD provides fast BFD peer failure detection times independently of all media types, encapsulations, topologies, and routing protocols [BGP](#), [EIGRP](#), [IS-IS](#), and [OSPF](#). By sending rapid failure detection notices to the routing protocols in the local router to initiate the routing table recalculation process, BFD contributes to greatly reduced overall network convergence time.

The BFD protocol has no discovery mechanisms to detect neighbors; it is designed solely as an agent for other applications requiring fast failure detection. Whenever a routing protocol that is configured to use BFD detects a new neighbor, it requests availability tracking from BFD.

BFD can rely on control packets or on echo packets. Echo packets are IP packets addressed to the router itself but sent to the Layer 2 address of the next-hop node. The echo packets thoroughly test the complete bidirectional forwarding path between adjacent routers because they have to be transmitted by the originating router, propagated to the adjacent router, received by its interface logic, switched by its forwarding engine, and sent back to the originator (because the IP packet is addressed to the router itself).

For example, when R1 sends a BFD echo packet, it sets the destination [IP address](#) in the packet to its own interface IP address and the [MAC address](#) in the Layer 2 frame header to the MAC address of the neighbor (R2). When R2 receives the packet, it performs a Layer 3 lookup and sends the packet toward its final destination (back to R1).



BFD operation:

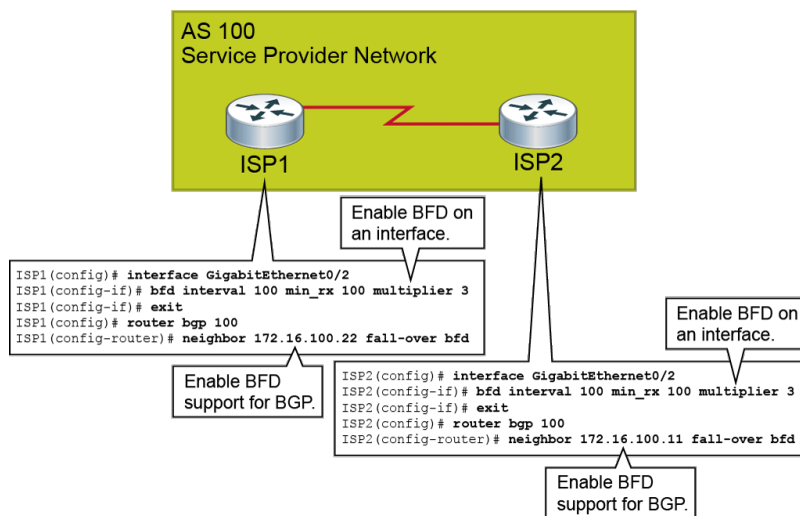
- Routing protocol (BFD client) bootstraps BFD to create a BFD session to a neighbor:
  - BFD client receives link status change notification.
  - Receive and transmit intervals are negotiated and configurable.
- The two systems agree on a method to detect failure.
- In case of failure, BFD notifies the BFD client:



– The BFD client independently decides on the action. When a routing protocol (BGP, for example) discovers a neighbor, it sends a request to the local BFD process to initiate a BFD neighbor session with the BGP neighbor router. Then the BFD neighbor session with the BGP neighbor router is established. If there is a failure on the link between neighbors, the BFD neighbor session with the BGP neighbor router is torn down. BFD notifies the local BGP process that the BFD neighbor is no longer reachable. The local BGP process tears down the BGP neighbor relationship. If an alternative path is available, the routers will immediately start converging on it.

**i** If multiple routing protocols want to establish BFD sessions with the same remote system for the same routed protocol (IPv4 or IPv6), all must share a single BFD session.

## BFD Configuration



To configure BFD, you first have to enable the BFD, using the **bfd interval** command in the interface configuration mode.

```
bfd interval send-timer min_rx receive-timer multiplier multiplier
```

### Syntax Description

| Parameter                           | Description                                                                                                                                                             |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>interval</b> <i>send-timer</i>   | The rate, in milliseconds, at which BFD control packets will be sent to BFD peers. The valid range for the <i>send-timer</i> is from 50 to 999.                         |
| <b>min_rx</b> <i>receiver-timer</i> | The rate, in milliseconds, at which BFD control packets will be expected to be received from BFD peers. The valid range for the <i>receive-timer</i> is from 50 to 999. |
| <b>multiplier</b> <i>multiplier</i> | The number of BFD packets that can be lost before the BFD peer is declared "down."                                                                                      |

Then you have to enable BFD support for BGP, using the **neighbor fall-over bfd** command.

```
neighbor ip-address fall-over bfd
```

### Syntax Description

| Parameter            | Description                          |
|----------------------|--------------------------------------|
| <b>fall-over bfd</b> | Enables BFD for individual neighbor. |

## BGP Nonstop Forwarding Awareness

Usually, when a networking device restarts, all routing peers of that device detect that the device went down and then came back up. This transition results in what is called a routing flap, which could spread across multiple routing domains. Routing flaps that are caused by routing restarts create routing instabilities, which are detrimental to the overall network performance. Cisco Nonstop Forwarding (NSF), also known as graceful restart, works with the [SSO](#) to minimize the amount of time a network is unavailable to its users following a switchover. The main objective of Cisco NSF is to continue forwarding IP packets following a route processor switchover in platforms with dual [RPs](#), thus reducing network instability.

### Cisco Nonstop Forwarding

- Cisco NSF is applicable in platforms with dual RPs and works together with SSO.
- Cisco NSF allows:
  - Routing neighbor relationships remain established during SSO.
  - Routes on neighboring routers remain valid.
  - Forwarding of data packets continues while the routing process on the new RP converges.
- Cisco NSF is supported by:
  - Routing protocols (OSPF, IS-IS, EIGRP, BGP)
  - Forwarding operation (Cisco Express Forwarding)
- The device must be Cisco NSF-capable.
- The neighboring device must be Cisco NSF-aware.

Cisco NSF allows for the forwarding of data packets to continue along known routes while the routing protocol information is being restored following a switchover. With [NSE](#), peer networking devices do not experience routing flaps. Data traffic is forwarded through intelligent line cards or dual forwarding processors, while the standby RP assumes control from the failed active RP during a switchover.

Cisco NSF is supported by four protocols for routing ([BGP](#), [EIGRP](#), [OSPF](#), and [IS-IS](#)), and by Cisco Express Forwarding (CEF) for forwarding. The routing protocols have been enhanced with Cisco NSF capability and awareness, meaning that routers running these protocols can detect a switchover and take the necessary actions to continue forwarding network traffic and to recover route information from the peer devices.

A networking device is said to be Cisco NSF-aware if it is running NSF-compatible software. A device is said to be Cisco NSF-capable if it has been configured to support NSF and can rebuild routing information from NSF-aware or NSF-capable neighbors.

Cisco NSF depends on CEF to continue forwarding packets during switchover while the routing protocols rebuild the [RIB](#) tables. Once the routing protocols have converged, CEF updates the [FIB](#) table and removes old route entries. CEF, in turn, updates the line cards with the new FIB information.

Cisco NSF overview:

- One RP is active, one is standby.
- Cisco Express Forwarding on the active RP synchronizes the FIB and adjacency table to the standby RP.
- Upon switchover, the new active RP uses the old FIB and adjacency table to forward packets while the routing protocol reconverges.
- Routing protocol must:
  - Establish neighbor relationship without causing a reset of neighbor relationship
  - Learn routing information
- As the routing protocol starts to repopulate the RIB, it updates Cisco Express Forwarding.

Cisco NSF always operates together with SSO. In specific Cisco networking devices that support dual RPs, SSO establishes one of the RPs as the active processor while the other RP is designated as the standby processor, and then synchronizes the FIB and adjacency table between them. A switchover from the active to the

standby processor occurs when the active RP fails, is removed from the networking device, or is manually taken down for maintenance. During the switchover, the new active RP uses the old FIB and adjacency table to forward packets while the routing protocol reconverges. Once the routing protocols have updated the RIB, CEF updates the FIB table and removes old route entries. CEF, in turn, updates the line cards with the new FIB information.

The routing protocols (for example, BGP) run only on the active RP, and they receive routing updates from their neighbor routers. Routing protocols do not run on the standby RP. If Cisco NSF is not configured, a neighboring device will tear down the adjacency during a switchover. The result would be a routing flap that would spread across the network.

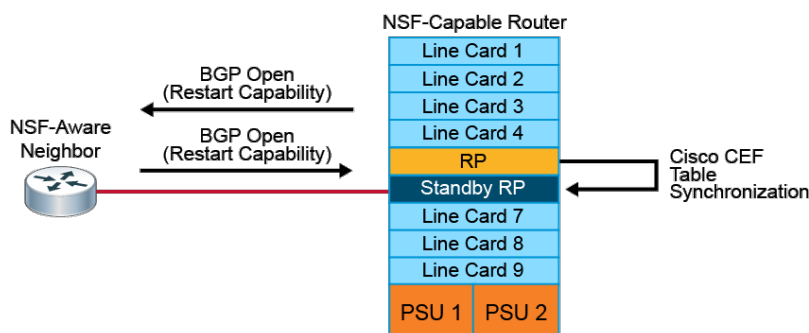
Using Cisco NSF, the routing protocols on an NSF-capable device requests that the NSF-aware neighbor devices send routing information to help rebuild the routing tables. The Cisco NSF-aware neighbor will not tear down the neighbor relationship. It will re-establish the neighbor relationship with the Cisco NSF-capable device, and will send routing information to synchronize the routing table on an NSF-capable device.

A Cisco NSF-aware router can peer with Cisco NSF-capable routers and facilitate the resynchronization of routing information with such routers.

## BGP Nonstop Forwarding Awareness

The BGP NSF Awareness feature allows the BGP peers of the failing router to retain the routing information that is advertised by the failing router. It also continue to use this information until the failed router has returned to normal operating behavior and is able to exchange routing information. The peering session is maintained throughout the entire NSF operation.

- Minimizes the effects of the following:
  - Well-known failure conditions (for example, a stuck-in-active event)
  - Unexpected events (for example, an RP switchover operation)
  - Scheduled events (for example, a hitless software upgrade)



```
router(config-router)# bgp graceful-restart
```

- Globally enables BGP NSF awareness.

The BGP NSF feature provides an NSF-aware router with the capability to detect a neighbor that is undergoing an SSO operation, maintain the peering session with this neighbor, retain known routes, and continue to forward packets for these routes.

BGP support for Cisco NSF requires that neighbor routers are NSF-aware or NSF-capable. The graceful restart mechanism also enables Cisco NSF awareness in BGP. A router that is NSF-aware functions like a router that is NSF-capable, with one exception: An NSF-aware router cannot perform an SSO operation. However, a router that is NSF-aware can maintain a peering relationship with a NSF-capable neighbor during an NSF SSO operation. It can hold routes for this neighbor during the SSO operation.



An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in an NSF restart operation.

The deployment of BGP NSF awareness can minimize the effects of the following:

- Well-known failure conditions (for example, a stuck-in-active event)
- Unexpected events (for example, an RP switchover operation)
- Scheduled events (for example, a hitless software upgrade)

BGP NSF improves the overall network stability by reducing the amount of resources that are normally required for reestablishing peering with a failed router.

When a Cisco NSF-capable router begins a BGP session with a BGP peer, it sends an OPEN message to the peer. Included in the message is a declaration that the Cisco NSF-capable or NSF-aware router has "graceful restart capability." If the BGP peer has received this capability, it is aware that the device sending the message is Cisco NSF-capable. Both the Cisco NSF-capable router and its BGP peers (NSF-aware peers) need to exchange the graceful restart capability in their OPEN messages at the time of session establishment. If both peers do not exchange the graceful restart capability, the session will not be capable of graceful restart.

If the BGP session is lost during the RP switchover, the Cisco NSF-aware BGP peer marks all the routes that are associated with the NSF-capable router as stale. However, it continues to use these routes to make forwarding decisions for a time. This functionality means that no packets are lost while the newly active RP is waiting for convergence of the routing information with the BGP peers.

After an RP switchover occurs, the Cisco NSF-capable router re-establishes the session with the BGP peer. In establishing the new session, it sends a new graceful restart message that identifies the Cisco NSF-capable router as having restarted.

At this point, the routing information is exchanged between the two BGP peers. Once this exchange is complete, the Cisco NSF-capable device uses the routing information to update the RIB and the FIB with the new forwarding information. The Cisco NSF-aware device uses the network information to remove stale routes from its BGP table. Once the stale routes are removed, the BGP protocol is fully converged.

If a BGP peer does not support the graceful restart capability, it will ignore the graceful restart capability in an OPEN message but will establish a BGP session with the Cisco NSF-capable device. This functionality will allow interoperability with non-NSF-aware BGP peers (and without NSF functionality), but the BGP session with non-NSF-aware BGP peers will not be capable of graceful restart.

BGP NSF awareness is not enabled by default. To globally enable it, use **bgp graceful-restart** router configuration command. To disable the BGP graceful restart capability globally for all BGP neighbors, use the **no** form of this command.

```
bgp graceful-restart [restart-time seconds] [stalepath-time seconds]
```

### Syntax Description

| Parameter                            | Description                                                                                                                                                                                                                                                                     |
|--------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>restart-time</b> <i>seconds</i>   | (Optional) The maximum time period that the local router will wait for a graceful-restart-capable neighbor to return to normal operation after a restart event occurs. The default value for this argument is 120 seconds. The valid range of values is from 1 to 3600 seconds. |
| <b>stalepath-time</b> <i>seconds</i> | (Optional) The maximum time period that the local router will hold stale paths for a restarting peer. All stale paths are deleted after this timer expires. The default value for this argument is 360 seconds. The valid range of values is from 1 to 3600 seconds             |



NSF awareness is enabled automatically in supported software images for Interior Gateway Protocols, such as EIGRP, IS-IS, and OSPF.

## BGP Scan Time

[BGP](#) convergence can also be reduced by using certain BGP timers. BGP scan time is one of the timers that can be used to influence BGP convergence speed.

- Defines how often BGP scanner process scans the BGP table.
- If lowered, can improve convergence
- Needed to confirm that next hops are still available.
- The BGP scanner process is also responsible for advanced features such as conditional advertisement check and performing route dampening.
- Set to 60 seconds by default.

```
router(config-router)# bgp scan-time scanner-interval
```

- Changes the default value of BGP scanner process runs.

The BGP scanner process walks (scans) the BGP table and confirms the reachability of next hops. A change of this status triggers a new BGP route selection for the network. The router then propagate the changes to established BGP neighbors. Increasing the BGP scanner process frequency will make the router find a changed status more quickly, but it will also consume more [CPU](#) resources.

The BGP scanner process is also responsible for some advanced BGP features. It checks the conditional advertisement to determine whether BGP should advertise conditional prefixes or perform route dampening.

To configure nondefault time interval for repetitions of the BGP scanner process, use **bgp scan-time** command.

```
bgp scan-time [import] scanner-interval
```

### Syntax Description

| Parameter               | Description                                                                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>import</b>           | (Optional) Configures import processing of <a href="#">VPNv4</a> unicast routing information from BGP routers into routing tables.                                                                            |
| <i>scanner-interval</i> | Specifies the scanning interval of BGP routing information. Valid values that are used for selecting the desired scanning interval are from 5 to 60 seconds. By default, the scanning interval is 60 seconds. |

## Monitoring BGP Scan Time

You can check the configured BGP scan interval using the **show ip bgp summary** command.

```
R1# show ip bgp summary
BGP router identifier 192.168.12.1, local AS number 1
BGP table version is 87, main routing table version 87
11 network entries using 1628 bytes of memory
16 path entries using 1024 bytes of memory
3/2 BGP path/bestpath attribute entries using 408 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3084 total bytes of memory
BGP activity 44/33 prefixes, 84/68 paths, scan interval 60 secs
<... output omitted ...>
```

- Scan interval is defined per BGP router process and address family.

The configured BGP scan interval will apply to the entire BGP routing protocol process.

## BGP Next-Hop Tracking

BGP monitors the next hop of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. By default, the BGP scanner is used to poll the [RIB](#) for this information every 60 seconds. During the 60 second time period between scan cycles, [IGP](#) instability or other network failures can cause black holes and routing loops to temporarily form.

The BGP next-hop tracking feature prevents black hole routing during the 60 second period of scan-timer. Scan-timer is already mostly replaced in basic functions by the BGP [NHT](#).

- The replacement for scan-timer.
  - Prevents black-hole routing during the scan-timer period.
- It is event driven.
- Automatically tracks BGP prefixes, and rapidly report next-hop changes to the BGP routing process.
- Enabled by default.

```
router(config-router)# bgp nexthop trigger [delay seconds | enable]
```

- Controls the BGP NGT feature:
  - Enables it, if it was previously disabled.
  - Changes the interval between checks.

BGP next-hop tracking is an event-driven system that provides faster convergence in on-demand fashion rather than periodic based on scan-timer. It monitors RIB for next-hop related changes for both [EBGP](#) and [IBGP](#) prefixes. It reports changes to the BGP routing process.

This optimization improves overall BGP convergence by reducing the response time to next-hop changes for routes installed in the RIB. When a best path calculation is run in between BGP scanner cycles, only next-hop changes are tracked and processed.

BGP NHT feature is enabled by default. Disabling next hop address tracking may be useful if your network has unstable IGP peers and route dampening is not resolving the stability issues. To control this feature, use the **bgp nexthop trigger** command in the address family configuration mode of the routing process.

```
bgp nexthop trigger { delay seconds | enable}
```

**Syntax Description**

| Parameter                   | Description                                                                                                                                                                                   |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>delay</b> <i>seconds</i> | Changes the delay interval between checks on updated next-hop routes that are installed in the routing table. Valid values for the delay are from 0 to 100 seconds. The default is 5 seconds. |
| <b>enable</b>               | Enables BGP next-hop tracking.                                                                                                                                                                |

The **bgp scan-time** command is ignored if your router has BGP next-hop tracking enabled for the address-family.

## BGP Advertisement Interval

[BGP](#) advertisement interval is another option to influence BGP convergence speed.

- Defines a time which has to elapse between two successive updates about the same destination that are sent to a neighbor.
- If lowered, can improve convergence
- Default values are different for IBGP and EBGP neighbors:
  - 30 seconds for EBGP neighbors
  - 0 seconds for IBGP neighbors

```
router(config-router)# neighbor [ip-address | peer-group-name] advertisement-interval seconds
```

- Changes the default time interval in the sending of BGP routing updates for a specific neighbor:

Advertisement interval timer controls the rate at which successive advertisements are sent to a BGP neighbor. When a BGP-speaking router sends a route update to a neighbor for a specific destination, it is not allowed to send another update to the neighbor about the same destination until a time equal to the advertisement interval has elapsed. So, the advertisement interval timer acts as a form of rate limiting on a per-destination basis, even though the value of the advertisement interval is configured for each neighbor.

The default values are different for [IBGP](#) and [EBGP](#) peers. For IBGP peers, the interval is set to 0 seconds. For EBGP peers, the interval is set to 30 seconds. For EBGP peers that are configured in a [VRF](#) instance, the interval is also set to 0 seconds.

It is important to note that while the router is waiting for the advertisement interval timer to expire, the router can still receive and process route updates from BGP neighbors. Changing the advertisement interval does not rate-limit BGP route selection (inbound updates and subsequent processing) but only the rate of outgoing route advertisements.

When faster propagation of successive BGP updates (which are batched and rate-limited) is required, the network administrator can lower the default value of the advertisement interval, thus improving convergence.

You can modify the default advertisement interval for a specific BGP peer using the **neighbor advertisement-interval** router configuration command.

```
neighbor [ip-address | peer-group-name] advertisement-interval seconds
```

### Syntax Description

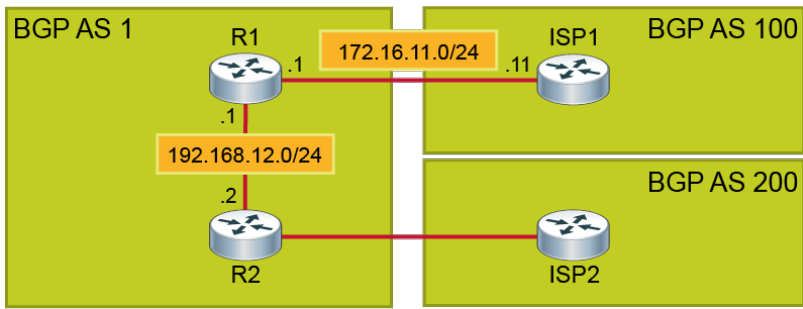
| Parameter              | Description                                                                                                                                                                                                  |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>      | Neighbor <a href="#">IP address</a> .                                                                                                                                                                        |
| <i>peer-group-name</i> | Name of a BGP peer group. If a BGP peer group is specified by using the <i>peer-group-name</i> argument, all members of the peer group will inherit the characteristic that is configured with this command. |
| <i>seconds</i>         | Time in seconds. The valid value is from 0 to 600 seconds. The default is 30 seconds for external peers and 5 seconds for internal peers.                                                                    |



In routers that handle large BGP tables and less stable networks, lowering the advertisement interval could potentially lead to consuming large portions of router resources.

## Monitoring the BGP Advertisement Interval

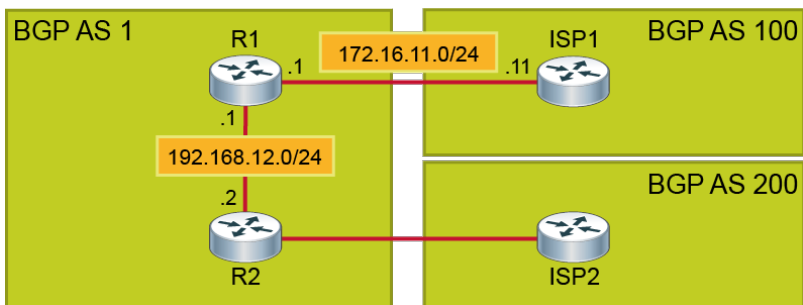




```

R1# show ip bgp neighbors 192.168.12.2
BGP neighbor is 192.168.12.2, remote AS 1, internal link
<... output omitted ...>
Default minimum time between advertisement runs is 0 seconds
<... output omitted ...>

```



```

R1# show ip bgp neighbors 172.16.11.11
BGP neighbor is 172.16.11.11, remote AS 100, external link
<... output omitted ...>
Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
 Session: 172.16.11.11
 BGP table version 102, neighbor version 102/0
<... output omitted ...>
Minimum time between advertisement runs is 15 seconds
<... output omitted ...>

```

You can examine the currently configured BGP advertisement interval with the **show ip bgp neighbors** command. The advertisement interval is defined for a specific neighbor in a specific BGP address family. Actual values of the advertisement interval are therefore stated under the specific address-family portion of the neighbor output.

The example shows the output for internal neighbor, where the default timer is 0 seconds. The second example shows the changed default timer for EBGp, 30 seconds. After the parameter was changed, the timer switched to 15 seconds.



## BGP Keepalive and Hold-Down Timers

BGP keepalive and hold-down timers are two BGP timers that influence BGP convergence as well.

BGP keepalive timer:

- Defines a time between successive keepalive messages.
- Set to 60 seconds by default.

BGP hold-down timer:

- Defines how long a router will wait from the last received keepalive or update message before declaring the session dead.
- Set to 3x keepalive = 180 seconds by default.

```
router(config-router)# neighbor [ip-address | peer-group-name] timers keepalive holdtime [min-holdtime]
```

The keepalive timer defines how often a BGP router will probe a BGP neighbor by sending keepalive messages. The hold down timer defines how long a router will wait from the last received keepalive or update message before declaring the session dead.

Lowering the keepalive and hold-down timers will improve BGP convergence.

By default, the keepalive timer is set to 60 seconds, and the hold-down timer is set to 180 seconds. Even if the IP routing table indicates that the neighbor is no longer reachable, BGP will not trigger the convergence process until the BGP session hold-down timer expires.

To configure a nondefault value of the keepalive and/or hold-down timers, use the **neighbor timers** router configuration command.

```
neighbor [ip-address | peer-group-name] timers keepalive holdtime [min-holdtime]
```

### Syntax Description

| Parameter              | Description                                                                                                                                                                                                                                                            |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>      | Neighbor <a href="#">IP address</a> .                                                                                                                                                                                                                                  |
| <i>peer-group-name</i> | Name of a BGP peer group. If a BGP peer group is specified by using the <i>peer-group-name</i> argument, all members of the peer group will inherit the characteristic that is configured with this command.                                                           |
| <i>keepalive</i>       | Time in seconds with which the keepalive messages are sent to the peer. The valid value is from 0 to 65535 seconds. The default is 60 seconds.                                                                                                                         |
| <i>holdtime</i>        | Time in seconds after not receiving a keepalive message and declaring a peer dead. The valid value is from 0 to 65535 seconds. The default is 180 seconds.                                                                                                             |
| <i>min-holdtime</i>    | (Optional) Interval (in seconds) specifying the minimum acceptable hold-time from a BGP neighbor. The minimum acceptable hold-time must be less than, or equal to, the interval that is specified in the <i>holdtime</i> argument. The valid value is from 0 to 65535. |



New keepalives will take effect only after new BGP session establishment as they are transferred and negotiated to the lower values from the two peers in the OPEN messages. Therefore you need to reset the BGP session.

## Monitoring BGP Keepalive and Hold-Down Timers

```
R1# show ip bgp neighbors
BGP neighbor is 172.16.11.11, remote AS 100, external link
BGP version 4, remote router ID 10.0.1.81
BGP state = Established, up for 06:54:31
Last read 00:00:02, last write 00:00:22, hold time is 180, keepalive int
```

```
erval is 60 seconds
<... output omitted ...>
```

- Verifies keepalive and hold-down timers.

You can examine the currently configured BGP keepalive and hold-down timers for each neighbor with the **show ip bgp neighbors** command.

In the example, both timers are left on default: keepalive timer is 60 seconds and hold-down timer is 180 seconds.

## Summary

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

- BGP is considered converged when there are no outstanding updates to be sent to neighbors.
- Several features are provided in BGP to improve convergence time:
  - **PMTU discovery** benefits BGP as it benefits any transmission over IP.
  - **Input queue depth** can be increased to improve performance.
  - **PIC** enhances BGP convergence regardless of number of BGP prefixes.
  - **BFD** reduces convergence time by providing rapid detection of failed peers.
  - **NSF awareness** reduces the amount of resources by continue forwarding packets during a failure condition.
- You can also lower the BGP timers to influence BGP convergence:
  - Scan timer
  - Advertisement interval
  - Keepalive and hold-down timers

# Limiting the Number of Prefixes Received from a BGP Neighbor

## Overview

There are currently more than 600,000 prefixes on the Internet. There are many circumstances in which network administrators do not need or desire their routers to carry full Internet routing. Furthermore, there is a need to provide protective controls on customer-facing routers to ensure that a configuration error does not cause the accidental advertisement of prefixes from [ASs](#) that did not originate them.

[BGP](#) is designed for reliability and scalability. As such, it has a tremendous amount of flexibility regarding administrative policy controls, route selection, and performance tuning and scalability features. An advanced BGP configuration tool has been designed to improve BGP scalability and performance by reducing the number of prefixes that a router receives from a BGP neighbor.

Upon completing this lesson, you will be able to:

- Describe the need for limiting the number of routes that are received from a BGP neighbor
- Identify the Cisco IOS command that is required to configure the BGP route limiting
- Describe how to configure and monitor BGP maximum-prefix function

## BGP Route Limiting

Incoming route policies, which are applied on BGP routers, indicate the BGP attribute values that a route should have in order to be accepted. Route policies can be applied that match routes based on the network number or the BGP attributes that are attached to a route. The most commonly applied filtering used in route policies is the one that matches the contents of the AS path attribute.

- All filtering mechanisms specify only what you are willing to accept but not how much.
- A misconfigured BGP neighbor can send a huge number of prefixes that can exhaust the memory of a router or overload the CPU (several Internet-wide incidents have already occurred).
- BGP maximum prefixes limiting is used to establish a hard limit on the number of prefixes that are received from a neighbor.

An ISP with a multihomed customer may use route policies to ensure that the routes that are received from the customer originate within the AS of the customer. Using an AS path access list is one method of achieving this goal.

If you configure route policies in a customer router improperly, it may accidentally cause the customer to receive many Internet routes. Even worse, a faulty configuration may cause customer to advertise prefixes as though the routes originated inside the customer AS. This situation would result in a BGP table in the ISP router that lists many of the possible destination networks on the Internet as reachable in the customer AS. The AS would contain only a single entry, the customer AS. The BGP route selection in the ISP network would select those routes as the best, based on the AS path length. As a result, it would direct much of the provider traffic that is intended for the Internet to the customer network.

A route policy that is based on an AS path filter in the ISP router will not prevent this accident. The routes that the customer sends have the anticipated AS path value. A route policy that is based on customer prefixes and that distinctly identifies and permits each of the network numbers that the customer may advertise will prevent the accident. However, such a routing policy is difficult to maintain.

BGP router terminates peering when a number of maximum prefixes exceeded.

You can configure a router to:

- Generate a logging message when a specified percentage of the maximum prefixes is reached.
- Reestablish BGP peering after a specified time (from 1 to 65535 minutes).
- Generate a logging message when the maximum prefix limit is exceeded, instead of terminating BGP peering.

A scalable solution to the need for limiting the number of routes (prefixes) that are received from a BGP neighbor is to use BGP maximum prefixes limiting. This feature enables you to specify how many routes a router can receive from a neighbor.

When the number of prefixes that is received from the peer for a given address family exceeds the maximum limit, a cease notification message is sent to the neighbor. Also the peering with the neighbor is terminated

You can also configure the router to generate a logging message when a configured percentage of the maximum prefixes is reached. You can also configure the router to automatically reestablish a peering session that has been disabled because the maximum prefix limit has been exceeded. Instead of ceasing the peering with a neighbor, you can configure the router to generate and display a warning when a configured percentage of the maximum prefixes has been reached.

## Configuring the BGP Route Limiting

The [BGP](#) route limiting can be configured with the use of maximum-prefix function.

```
router(config-router)# neighbor ip-address maximum-
prefix maximum [threshold] [warning-only] [restart restart-interval]
```

- This command controls how many prefixes can be received from a neighbor.
  - The optional *threshold* parameter specifies the percentage where a warning message is logged (default is 75%).
  - The optional **warning-only** keyword specifies the action on exceeding the maximum number (default is to drop the neighbor relationship).
  - The optional **restart** keyword instructs the router to try to re-establish the session after the specified interval in minutes.

To control how many prefixes a BGP router can receive from a neighbor, use the **neighbor maximum-prefix** router configuration command.

```
neighbor {ip-address| peer-group-name} maximum-
prefix maximum [threshold] [warning-only] [restart restart-interval]
```

### Syntax Description

| Parameter              | Description                                                                                                                                                                                                                            |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>      | IP address of the neighbor.                                                                                                                                                                                                            |
| <i>peer-group-name</i> | Name of a BGP peer group.                                                                                                                                                                                                              |
| <i>maximum</i>         | Maximum number of prefixes that are allowed from this neighbor.                                                                                                                                                                        |
| <i>threshold</i>       | (Optional) Integer specifying at what percentage of maximum the router starts to generate a warning message. The range is 1 to 100 percent. The default is 75 percent.                                                                 |
| <b>warning-only</b>    | (Optional) Allows the router to generate a log message when the maximum is exceeded, instead of terminating the peering.                                                                                                               |
| <b>restart</b>         | (Optional) Configures the router to automatically re-establish a peering session that has been disabled because the maximum-prefix limit has been exceeded. The configurable range of the restart interval is from 1 to 65535 minutes. |

This command allows you to configure a maximum number of prefixes that a BGP router is allowed to receive from a peer. It adds another mechanism (in addition to distribute lists, filter lists, and route maps) to control prefixes that are received from a peer.

When the number of received prefixes exceeds the maximum number that is configured, the router terminates the peering (by default). However, if the **warning-only** keyword is configured, the router sends a log message but continues peering with the sender. If the peer is terminated, the peer session remains down until the **clear ip bgp** command is issued on the router, unless you have included the restart keyword in the configuration.

The BGP Restart Session After Max-Prefix Limit feature enhances the capabilities of the neighbor **maximum-prefix** command with the introduction of the **restart** keyword. This enhancement allows you to configure the time interval after which a router re-establishes a peering session when the number of prefixes that have been received from a peer has exceeded the maximum prefix limit. The **restart** keyword has a configurable timer argument that is specified in minutes. The time range of the timer argument is from 1 to 65535.

This feature attempts to re-establish a disabled peering session at the time interval that you configure. However, the configuration of the restart timer alone cannot change or correct a peer that is sending an excessive number of prefixes. You will need to reconfigure the maximum-prefix limit or reduce the number of prefixes that are sent from the peer. A peer that is configured to send too many prefixes can cause instability in the network, where an excessive number of prefixes are rapidly advertised and withdrawn. In this case, the **warning-only** keyword can be configured to disable the restart capability while you correct the underlying problem.

You can use the **bgp dampening** command to configure

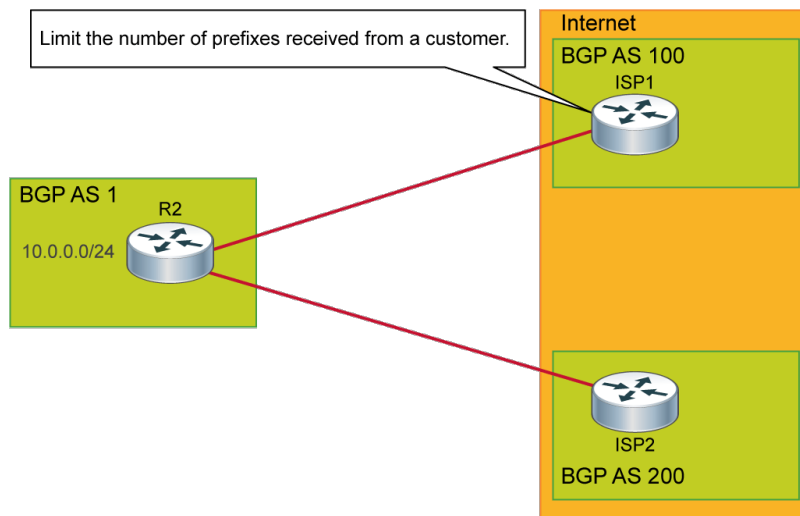


the dampening of a flapping route or interface when a peer is sending too many prefixes and causing network instability. You should need the **restart** command only when you are troubleshooting or tuning a router that is sending an excessive number of prefixes.

## Discovery 18: Configure BGP Route Limiting

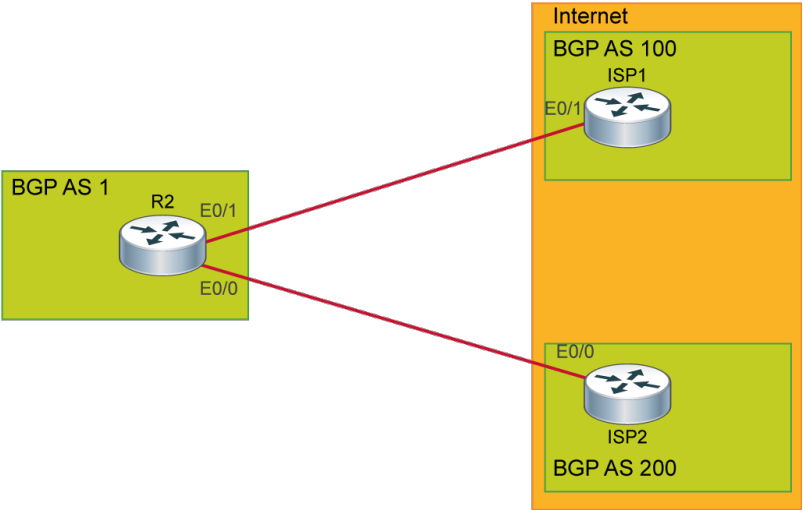
### Overview

Through this discovery, you will learn how to configure [BGP](#) route limiting using the maximum-prefix function. You will configure a service provider's router, ISP1, to limit the maximum number of prefixes that are received from a multihomed customer's router, R2.





Topology



## Job Aids

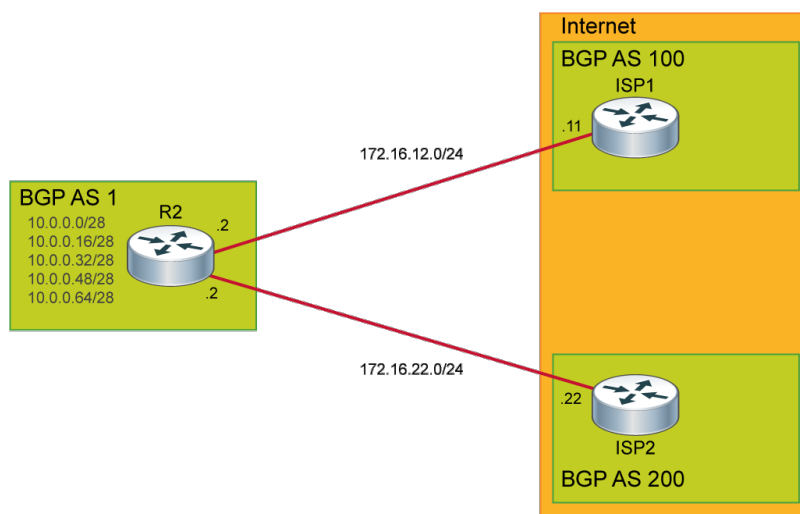


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, IOL behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

### Device Information

| Device | Interface                                                                                                                  | IP address                                                                                                                                    | Description                     |
|--------|----------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| R2     | Ethernet 0/0                                                                                                               | 172.16.22.2/24                                                                                                                                | Connection to ISP2              |
| R2     | Ethernet 0/1                                                                                                               | 172.16.12.2/24                                                                                                                                | Connection to ISP1              |
| R2     | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4<br>Loopback 5<br>Loopback 6<br>Loopback 7<br>Loopback 8<br>Loopback 9 | 10.0.0.1/28<br>10.0.0.17/28<br>10.0.0.33/28<br>10.0.0.49/28<br>10.0.0.65/28<br>10.0.0.81/28<br>10.0.0.97/28<br>10.0.0.113/28<br>10.0.0.129/28 | Loopbacks simulate LAN networks |
| ISP1   | Ethernet 0/1                                                                                                               | 172.16.12.11/24                                                                                                                               | Connection to R2                |
| ISP1   | Ethernet 0/2                                                                                                               | 172.16.100.11/24                                                                                                                              | Connection to ISP2              |
| ISP2   | Ethernet 0/0                                                                                                               | 172.16.22.22/24                                                                                                                               | Connection to R2                |
| ISP2   | Ethernet 0/2                                                                                                               | 172.16.100.22/24                                                                                                                              | Connection to ISP1              |

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#):

- R2 to ISP1
- R2 to ISP2

## Configure BGP Route Limiting

### Discovery Steps

#### Step 1

On the ISP1 and ISP2 routers, verify initial state of BGP table. Verify that you received routes to four destination networks from R2.

Use the **show ip bgp** command.

```
ISP1# show ip bgp
BGP table version is 5, local router ID is 172.16.12.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/28 172.16.12.2 0 0 1 i
*> 10.0.0.16/28 172.16.12.2 0 0 1 i
*> 10.0.0.32/28 172.16.12.2 0 0 1 i
*> 10.0.0.48/28 172.16.12.2 0 0 1 i
*> 10.0.0.64/28 172.16.12.2 0 0 1 i
```

```
ISP2# show ip bgp
BGP table version is 5, local router ID is 172.16.22.22
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/28 172.16.22.2 0 0 1 i
*> 10.0.0.16/28 172.16.22.2 0 0 1 i
*> 10.0.0.32/28 172.16.22.2 0 0 1 i
*> 10.0.0.48/28 172.16.22.2 0 0 1 i
*> 10.0.0.64/28 172.16.22.2 0 0 1 i
```

Both, ISP1 and ISP2, have received routes to five different networks—10.0.0.0/28, 10.0.0.16/28, 10.0.0.32/28, 10.0.0.48/28, and 10.0.0.64/24.

#### Step 2

On the ISP1 router, limit the number of received prefixes from R2.

ISP1 should receive no more than 8 prefixes from R2, and should start generating warning messages when the number of received prefixes exceeds 75% of the maximum specified.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.12.2 maximum-prefix 8 75
```

#### Step 3

On the R2 router, advertise the 10.0.0.80/28 and 10.0.0.96/28 networks in BGP.

To advertise networks, configure the following commands on R2.

**NOTE:** Before configuring R2, make sure that you also have a console window open for ISP1 router. So you will be able to see the generated warning messages.

```
R2(config)# router bgp 1
R2(config-router)# network 10.0.0.80 mask 255.255.255.240
R2(config-router)# network 10.0.0.96 mask 255.255.255.240
```

#### Step 4

On the ISP1 router, observe the warning message generated.

```
ISP1#
*Apr 2 12:07:55.964: %BGP-4-
MAXPFX: Number of prefixes received from 172.16.12.2 (afi 0) reaches 7, max 8
```

The warning message was generated, because ISP1 received 7 prefixes from R2. The maximum is set to 8, so the threshold of 75% was exceeded.

### Step 5

On the ISP1 router, verify that you still have all destination, that R2 advertises, in the BGP table.

```
ISP1# show ip bgp
BGP table version is 8, local router ID is 172.16.12.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/28 172.16.12.2 0 0 1 i
*> 10.0.0.16/28 172.16.12.2 0 0 1 i
*> 10.0.0.32/28 172.16.12.2 0 0 1 i
*> 10.0.0.48/28 172.16.12.2 0 0 1 i
*> 10.0.0.64/28 172.16.12.2 0 0 1 i
*> 10.0.0.80/28 172.16.12.2 0 0 1 i
*> 10.0.0.96/28 172.16.12.2 0 0 1 i
```

Although the warning was generated, ISP1 still has all 7 networks that are received from R2 in its BGP table.

### Step 6

On the R2 router, also advertise the 10.0.0.112/28 and 10.0.0.128/28 networks in BGP.

To advertise networks, configure the following commands on R2.

**NOTE:** Before configuring R2, make sure that you also have a console window open for ISP1 router. So you will be able to see the generated messages.

```
R2(config)# router bgp 1
R2(config-router)# network 10.0.0.112 mask 255.255.255.240
R2(config-router)# network 10.0.0.128 mask 255.255.255.240
```

### Step 7

On the ISP1 router, observe the generated messages.

```
ISP1#
*Apr 2 12:22:01.859: %BGP-4-MAXPFX: Number of prefixes received from 172.16.12.2 (afi 0) reaches 8, max 8
ISP1#
*Apr 2 12:22:32.575: %BGP-3-
MAXPFXEXCEED: Number of prefixes received from 172.16.12.2 (afi 0): 9 exceeds limit 8
*Apr 2 12:22:32.575: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Down BGP Notification sent
*Apr 2 12:22:32.575: %BGP-3-NOTIFICATION: sent to neighbor 172.16.12.2 3/1 (update malformed) 0 bytes
*Apr 2 12:22:32.575: %BGP-4-MSGDUMP: unsupported or mal-formatted message received from 172.16.12.2:
FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF 0037 0200 0000 1B40 0101 0040 0206 0201
0000 0001 4003 04AC 100C 0280 0404 0000 0000 1C0A 0000 80
ISP1#
*Apr 2 12:22:32.575: %BGP_SESSION-5-
ADJCHANGE: neighbor 172.16.12.2 IPv4 Unicast topology base removed from session BGP Notification sent
ISP1#
*Apr 2 12:23:41.141: %BGP-3-
NOTIFICATION: sent to neighbor 172.16.12.2 passive 2/8 (no supported AFI/SAFI) 3 bytes 000101 (timer expired)
```

The beginning of the output first shows that the total number of received prefixes has reached 8, which is over the threshold to generate a warning message. When you started to advertise the second network in BGP, the second message was generated. The total number of received prefixes is now 9, which is above the configured limit (8). As a result, the session with the neighbor, R2, was terminated.

### Step 8

On the ISP1 and ISP2 routers, verify the state of BGP table.

```
ISP1# show ip bgp
ISP1#
```

```
ISP2# show ip bgp
BGP table version is 10, local router ID is 172.16.22.22
```

```

Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

```

| Network          | Next Hop    | Metric | LocPrf | Weight | Path |
|------------------|-------------|--------|--------|--------|------|
| *> 10.0.0.0/28   | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.16/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.32/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.48/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.64/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.80/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.96/28  | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.112/28 | 172.16.22.2 | 0      |        | 0      | 1 i  |
| *> 10.0.0.128/28 | 172.16.22.2 | 0      |        | 0      | 1 i  |

ISP1 has no routes in the BGP table, since R2 was the only neighbor and the session was terminated due to exceeded maximum number of received prefixes. You did not configure route limiting on ISP2, so you can see routes for all nine destinations in its BGP table.

## Monitoring the BGP Route Limiting

```
router# show ip bgp neighbors [address]
```

- For neighbors with the maximum-prefix function configured, displays the maximum number of prefixes and the warning threshold.
- For neighbors exceeding the maximum number of prefixes, displays the reason that the BGP session is idle.

You can use the **show ip bgp neighbors** command to monitor the status of BGP neighbors. Among other things, the command displays information about how many prefixes a BGP router has received from a neighbor and if any limits have been configured.

If the peer exceeded the configured maximum number of prefixes, the output of the **show ip bgp neighbors** command shows the reason for resetting the session. As a result of the session being reset, the BGP session will remain in the Idle state.

To force the neighbor from the Idle state into the Active state and to re-establish the BGP session, you must issue the **clear ip bgp ip-address** command for the neighbor. Except, if you have specified the restart keyword in the configuration. In this case, the router tries to re-establish the BGP session automatically after the expiration of the configured restart timeout interval.

### Step 9

On the ISP1 router, verify information about the R2 neighbor.

To verify information about the neighbor, use the **show ip bgp neighbors** command with a neighbor IP address as an argument.

```

ISP1# show ip bgp neighbor 172.16.12.2
BGP neighbor is 172.16.12.2, remote AS 1, external link
BGP version 4, remote router ID 10.0.0.129
BGP state = Idle
Neighbor sessions:
 0 active, is not multisession capable (disabled)
Stateful switchover support enabled: NO
Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
BGP table version 17, neighbor version 1/17
Output queue size : 0
Index 0, Advertise bit 0
Address family not supported notification sent
Slow-peer detection is disabled
Slow-peer split-update-group dynamic is disabled
Peer had exceeded the max. no. of prefixes configured.
Maximum prefixes allowed 8
Threshold for warning message 75%
Reduce the no. of prefix and clear ip bgp 172.16.12.2 to restore peering
<... output omitted ...>

```

The state of BGP session with R2 router is Idle. The reason is that the peer had exceeded the maximum number of prefixes configured. You can see that the maximum allowed prefixes for this neighbor are 8 and the threshold for warning

messages is set to 75%. These are the values that you configured earlier.

To force the neighbor from the Idle state into the Active state and to re-establish the BGP session, you have to reduce the number of prefixes and clear the BGP session with the neighbor.

## Summary

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

- An improperly configured filter in a customer router may accidentally cause the customer router to receive many Internet routes.
- Maximum prefix limitation will shut down peering session if too many prefixes are received.
  - Use **neighbor maximum-prefix** command to configure route limiting—a maximum number of prefixes that a BGP router is allowed to receive from a peer.

# Implementing BGP Peer Groups

## Overview

Scaling routers to meet the demands of full Internet routing and associated administrative policies requires protocols like [BGP](#) with embedded scalability mechanisms. In environments where network administrators must configure many BGP peers, peer groups are a scalability tool that reduces both administrative overhead and router resource requirements.

Typical service provider networks usually contain BGP-speaking routers that consist of many neighbors that are configured with the same administrative policies. These policies can be outbound route maps, distribute lists, filter lists, update source, and so on. You can group neighbors with the same update policies into peer groups to simplify configuration and, more importantly, to make BGP updates more efficient. In this lesson, you will learn about peer groups as a BGP scalability mechanism.

Upon completing this lesson, you will be able to:

- Describe the need for BGP peer groups
- Describe the performance benefits of using BGP peer groups
- Describe the limitations of BGP peer groups
- Identify the Cisco IOS commands that are required to configure BGP peer groups
- Describe the configuration of BGP peer groups
- Explain the BGP peer group configuration examples
- Describe BGP dynamic update groups
- Describe BGP peer templates
- Describe the BGP peer template inheritance
- Identify the Cisco IOS commands that are required to configure BGP peer templates



## BGP Peer Groups Overview

In many cases, you as a network administrator must configure a single router with many neighbors, each neighbor having parameters similar to the others. This situation may cause time-consuming configuration work, because then you have to configure almost the same parameters for all of the neighbors. Imagine you have a service provider network that has an edge router with many customers attached to it, where each customer requires its own BGP session. You may find that all of the BGP sessions to its customer routers have almost identical configurations.

- BGP routers could have many neighbors with similar requirements:
  - Provider edge router with many customer connections.
  - BGP route reflector with many IBGP peers.
  - Provider edge router at an exchange point.
- Most of the parameters that are specified for the BGP neighbors are identical, with a few exceptions.

**Solution** is to group common parameters in a BGP peer group.

Likewise, IBGP sessions are almost always identically configured. If a full mesh is deployed within an AS, many peer configurations might exist. Recall that an AS containing only 15 routers will require  $([15 * 14] / 2) = 105$  neighbor sessions to meet the full-mesh requirement of BGP. Configuring 105 neighbors with duplicate parameters lead to a tremendous amount of redundant configuration.

To ease the burden of configuring many neighbors with identical or similar parameters (for example, route maps, filter lists, or prefix lists), the concept of peer groups was introduced. You as a network administrator can configure a template, or peer group with all the BGP parameters that are to be applied to many BGP peers. Actual BGP neighbors are bound to the peer group, and you apply the peer group configuration on each of the BGP sessions. BGP neighbors of a single router can be divided into several groups, each group having its own BGP parameters. Actual neighbors are then bound to the appropriate group, resulting in an optimum BGP configuration.

- BGP peer group creates a neighbor parameter template.
- Configurable parameters include the following:
  - Community propagation
  - Update-source and next-hop-self
  - EBGP multihop
  - Authentication password
  - Neighbor weight
  - Prefix lists, filter lists, and route maps
- Individual parameters that are specified in a peer group can be overridden on a neighbor-by-neighbor basis.

On a router, the peer group is created as a template. The template can be, among others, configured to do the following:

- Propagate, or not propagate, the community attribute.
- Use the IP address of a specific interface as the source address when opening the TCP session or use the next-hop-self feature.
- Use, or not use, the EBGP multihop function.
- Use, or not use, MD5 authentication on the BGP sessions.
- Filter out any incoming or outgoing routes using a prefix list, a filter list, and a route map.
- Assign a particular weight value to the routes that are received.

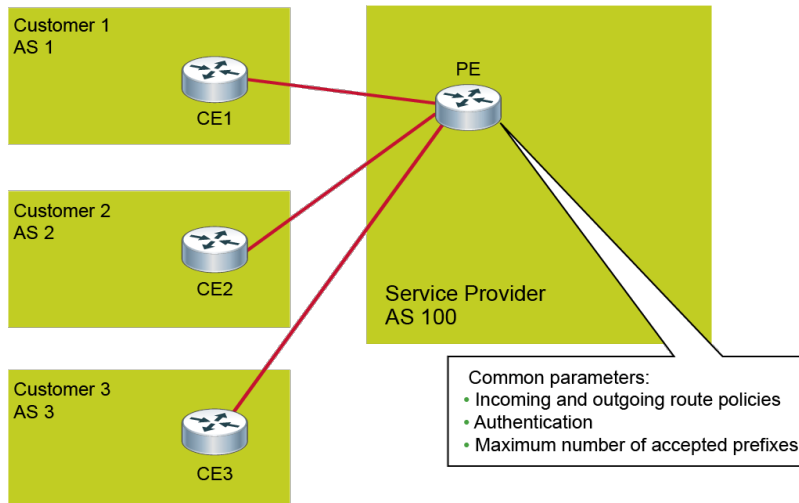
When actual neighboring routers are assigned to the peer group on a router, all of the attributes that are configured for the peer group are applied to all peer group members. Cisco IOS software optimizes the outgoing routes by running through the outgoing filters and routing policies only once and then replicating the results to each of the peer group members. In reality, Cisco IOS software assigns a peer group leader, for which the software generates an update, and this update is replicated by the leader to all other members of the peer group.

Some parameters that are configured on the peer group can be overridden by neighbor configurations. But only if the individual configurations apply on incoming updates. Outgoing updates are always prepared for the peer group leader and then

replicated to the other members of the peer group.

## Peer Groups Example—Customer Connections

This example illustrates a service provider network with a group of customer autonomous systems that can be treated in the same (or a very similar) way.



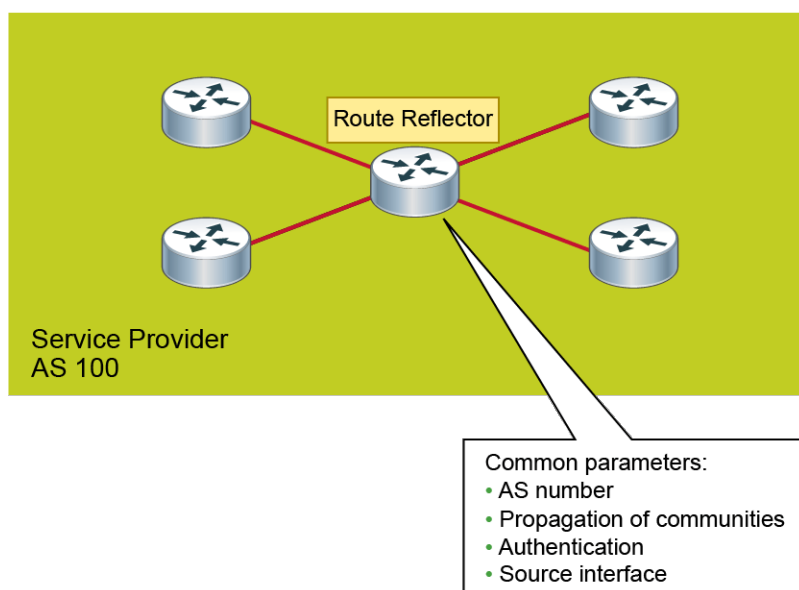
The figure shows an example where peer groups are useful. The customer autonomous systems are all assumed to announce local networks only. All customer ASs should receive BGP updates with the same set of Internet routes, and the customer ASs are all assumed to generate only a few prefixes.

This situation makes the neighbor configuration almost identical for each of the customers, with only a few changes that are specific to each neighbor.

In this scenario, the use of the peer group function is highly desirable. You as a network administrator can configure BGP neighbors in the customer ASs using a single peer group. You configure the peer group template with references to route policies, authentication settings, and the maximum number of received prefixes. Then the IP addresses of the customer routers are bound to the peer group, and the peer group configuration is applied to all neighbors.

## Peer Group Example—BGP Route Reflector

This example illustrates the BGP route reflector.



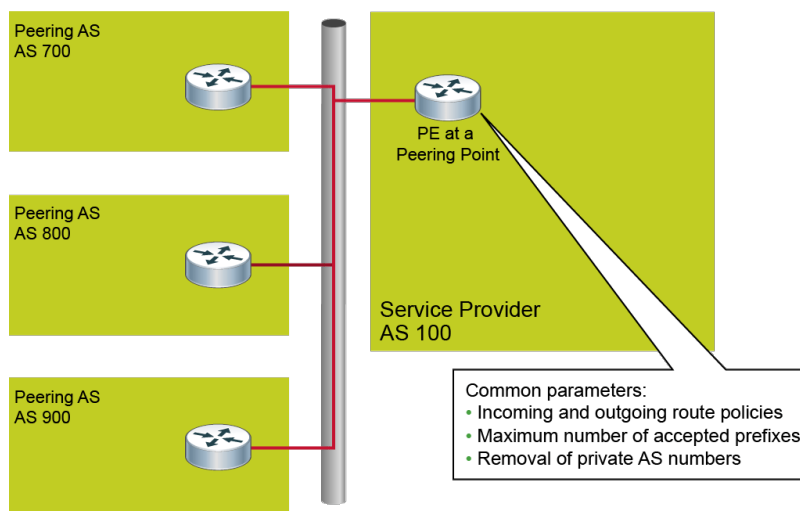
Another example of using peer groups is within the entire local AS, on the route reflectors, where every IBGP session is configured identically. If a router in the AS is supplied with some information, then all the routers should be supplied with the same information. Otherwise, an inconsistent routing policy within the AS might cause inconsistent routing or application of BGP policies.

The peer group function is a good tool to make sure that all IBGP peers receive the same configuration information. You can configure a peer group template with the required parameters. You can configure the neighbor AS number, enable of the **send-community** option, set of the update source to a loopback interface, and router authentication mechanisms. Then, all the internal neighbor IP addresses are bound to the peer group, and the peer group configuration is applied to all of them. This approach ensures a consistent routing policy within the AS.

In a service provider network, the routers that are assigned as route reflectors are the routers with the largest number of IBGP sessions. These are the routers where the peer group function is most useful.

## Peer Group Example—Edge Router at a Peering Point

This example illustrates an edge router at a peering point.



Another example of using peer groups is on the edge router that is located in the network where the service provider exchanges routes with other service providers. From the edge router, the service provider AS can peer with many other service providers.

All peering ASs should receive the same set of routes, namely the routes local to the service provider AS and the routes that are received from customer ASs. Also, all routes that are received by the service provider peering router from all peering ASs are processed almost identically. The characteristic of the exchange network is the same regardless of which neighbor the routes are received from. If the peering point is an [FDDI](#), [ATM](#), [Gigabit Ethernet](#), or [DPT](#) network, the preference of using the network for packet exchange may be different. However, for each single peering point, all neighbors are reachable over the same network, and the preference is quite likely to be the same.

Additionally, a number of other parameters could be the same, such as removing private AS numbers and limiting the number of routes received. In these cases, you can apply these parameters on the peer group template before the actual IP addresses of the neighbors are bound to the peer group.

## BGP Peer Groups as a Performance Tool

As described here, peer groups can be used to combine common [BGP](#) configuration into a peer group. Neighbors are then configured by assigning them to the peer group. By default, router builds BGP updates for each neighbor individually. Building BGP updates involves a number of router-[CPU](#)-consuming tasks, including scanning the BGP table and applying various outgoing filtering mechanisms (filter lists, route maps, and prefix lists). These tasks mean that when a router is configured with a large number of neighbors, the CPU load grows proportionally.

By default, router builds individual BGP updates for each BGP neighbor.

Common configuration can be combined into a peer group.

- Neighbors are configured by assigning them to the peer group.
- A single BGP update is then built for all members of a BGP peer group.
  - The CPU load does not increase linearly with the increased number of neighbors.
  - Use peer groups wherever possible to reduce the CPU load of the BGP process.

However, with the use of peer groups, some of the router CPU utilization that is imposed by BGP update generation is significantly reduced. The use of peer groups allows the router to run the BGP update (including all outgoing filter processing) only once for the entire peer group. The router, after it has finished building the BGP update, sends it to each member of the peer group. The actual [TCP](#) transmission still has to be done on a per-neighbor basis because of the connection-oriented characteristics of BGP sessions.

Router CPU load does increase when there are more neighbors of a router, due of increased TCP workload, but the use of peer groups can significantly reduce the increase. Therefore, you should use peer groups whenever possible to reduce the CPU load.



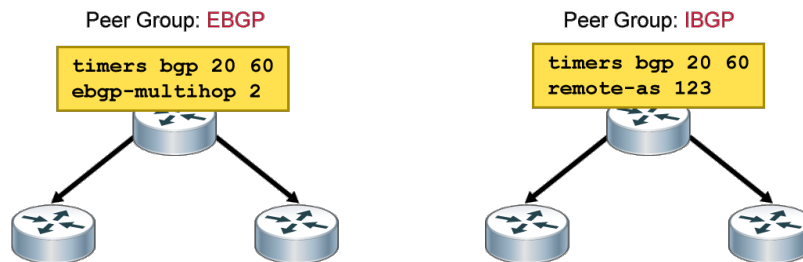
BGP peer groups are the fundamental BGP scalability tool and should be used in all environments where a router has a large number of BGP neighbors.



Routers built BGP updates for each neighbor individually before the introduction of dynamic update groups, which were introduced in Cisco IOS Release 12.0(24)S. Dynamic update groups, which are enabled by default, allow routers to dynamically calculate and optimize updates that are sent to neighbor that share the same outbound policies. Although this new feature effectively made manually defined peer groups obsolete, many administrators still use peer groups as they offer convenience during configuration.

## BGP Peer Group Limitations

BGP peer groups were introduced primarily for CPU usage optimization. Because the router builds only one update for all members of the same peer group, some restrictions apply to members of the peer group.



Peer groups have a number of limitations because of the way that they are used to build BGP updates.

- Peer groups were intended to be used only for CPU optimization.
- Awkward with similar but not identical configuration policies.
- IBGP and EBGP neighbors cannot be mixed in a peer group.
- Per-neighbor BGP parameters that affect outbound updates cannot be changed for peer group members.

BGP parameters and outbound routing policies cannot differ between group members because doing so could cause different updates to be sent to two members of the same peer group. Router creates only one update, which is then replicated to all members. Therefore, peer groups are awkward when used on a router with several neighbors that have similar but not completely identical policies.

Also EBGP and IBGP updates are very different. The same peer group cannot be used for EBGP and IBGP neighbors, even though a part of the configuration is the same for both types of neighbors. An example would be if you wanted to change the BGP keepalive and hold-down timers for both IBGP and EBGP sessions. Using peer groups, you must specify the timers in both peer groups (IBGP and EBGP), because they use different configurations due to the different nature of IBGP and EBGP sessions. Also EBGP updates have the AS path attribute changed. IBGP sessions pass only the local preference attribute. The MED attribute that is received from a remote AS is passed on to IBGP sessions but is removed before it is sent on an EBGP session. Therefore common BGP parameters must be replicated across all peer groups, and there is no way to use only one peer group for common configuration.

## Configuring BGP Peer Groups

To finish [BGP](#) peer group configuration, you first have to create a peer group, specify parameter for the peer group, and then assign neighbors into the peer group.

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

- Creates a BGP peer group (peer groups names are case-sensitive) .

```
router(config-router)# neighbor group-name any-BGP-parameter
```

- Specifies any BGP parameter for the peer group.

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

- Assigns a BGP neighbor to a peer group. The neighbor inherits all the BGP parameters that are specified for the peer group.

```
router(config-router)# neighbor ip-address any-BGP-parameter
```

- Overrides a BGP parameter that is specified for the peer group with a neighbor-specific parameter.

To configure BGP peer groups on Cisco IOS routers, perform the following steps:

1. **Create a BGP peer group.**

To create a BGP peer group, use the **neighbor peer-group** router configuration command.

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

2. **Specify parameters for the BGP peer group.**

After you have created a peer group, you can configure it with the **neighbor** commands. By default, members of the peer group inherit all the configuration options of the peer group. You can also configure members to override the options that do not affect outbound updates. Peer group members will always inherit the following configuration options: **remote-as** (if configured), **version**, **update-source**, **out-route-map**, **out-filter-list**, **out-dist-list**, **minimum-advertisement-interval**, and **next-hop-self**. All peer group members will inherit changes that are made to the peer group.

3. **Create a BGP neighbor.**

4. **Assign this neighbor into the peer group.**

To configure a BGP neighbor to be a member of a peer group, use the **neighbor peer-group** router configuration command.

```
neighbor ip-address peer-group peer-group-name
```

### Syntax Description

| Parameter              | Description                                                                                                 |
|------------------------|-------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>      | <a href="#">IP address</a> of the BGP neighbor that belongs to the peer group that is specified by the tag. |
| <i>peer-group-name</i> | Name of the BGP peer group to which this neighbor belongs.                                                  |

After you have assigned an actual neighbor to be a member of the peer group, all configurations that are made to the peer group template are then applied to all the neighbors that are assigned to that peer group. Through configuration, peer group configurations may be overridden for an individual neighbor, as long as the changes apply only to incoming updates. Remember that outgoing updates are prepared only once and then replicated.



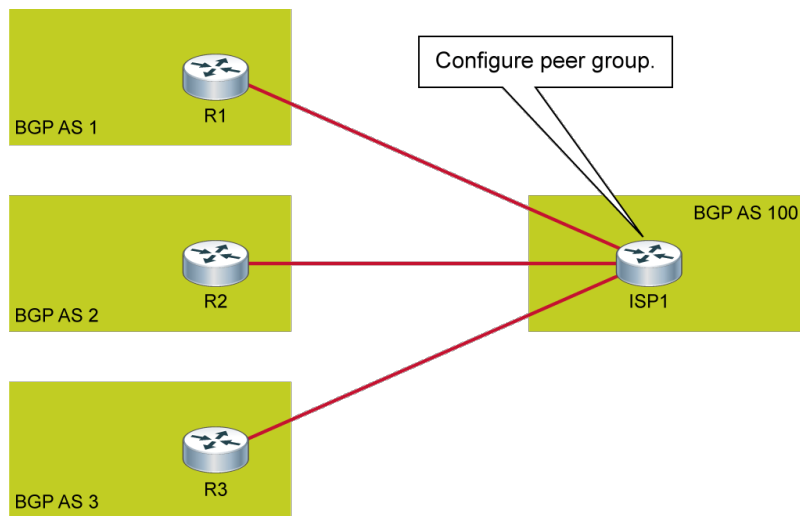
The peer group is a very powerful tool when network administrators are dealing with a large number of neighbors with almost identical configurations. However, if any of the customers require routing information that

differs from that of other members of the peer group, then that neighbor must be removed from the peer group and configured as an individual neighbor.

## Discovery 19: Configure BGP Peer Groups

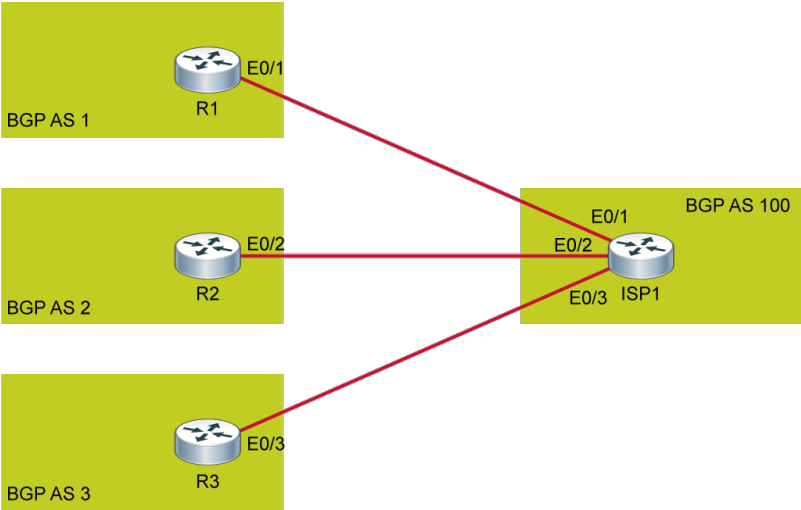
### Overview

Through this discovery, you will learn how to configure [BGP](#) peer group. A peer group will be used for all customer of the service provider because they share identical routing policy. You will put all neighbor configuration in the peer group configuration and the peers will then inherit the peer group configuration.





Topology



## Job Aids

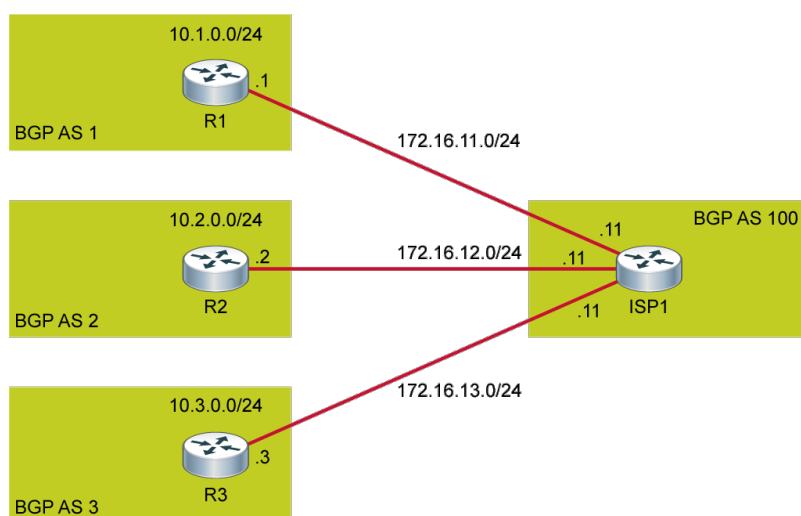


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

### Device Information

| Device | Interface    | IP address     | Description                                    |
|--------|--------------|----------------|------------------------------------------------|
| ISP1   | Ethernet 0/1 | 172.16.11.1/24 | Connection to R1                               |
| ISP1   | Ethernet 0/2 | 172.16.12.1/24 | Connection to R2                               |
| ISP1   | Ethernet 0/3 | 172.16.13.1/24 | Connection to R3                               |
| R1     | Ethernet 0/1 | 172.16.11.1/24 | Connection to ISP1                             |
| R1     | Loopback 1   | 10.1.0.1/24    | Loopback simulates <a href="#">LAN</a> network |
| R2     | Ethernet 0/2 | 172.16.12.2/24 | Connection to ISP1                             |
| R2     | Loopback 1   | 10.2.0.1/24    | Loopback simulates LAN network                 |
| R3     | Ethernet 0/3 | 172.16.13.3/24 | Connection to ISP1                             |
| R3     | Loopback 1   | 10.3.0.1/24    | Loopback simulates LAN network                 |

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#):

- ISP1 to R1
- ISP1 to R2
- ISP1 to R3

## Configure BGP Peer Groups

### Discovery Steps

#### Step 1

On the ISP1 router, verify that you receive networks, that all three customers advertise.

```
ISP1# show ip bgp
BGP table version is 4, local router ID is 172.16.13.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.1.0.0/24 172.16.11.1 0 150 0 1 i
*> 10.2.0.0/24 172.16.12.2 0 150 0 2 i
*> 10.3.0.0/24 172.16.13.3 0 150 0 3 i
```

You can see that ISP1 router in AS 100 received three networks, each from different customer:

- 10.1.0.0/24 network originates in the first customer's [AS 1](#).
- 10.2.0.0/24 network originates in the second customer's AS 2.
- 10.3.0.0/24 network originates in the third customer's AS 3.

Note that all three networks have local preference set to 150. Route map is used to set the local preference for all three neighbors.

```
ISP1# show route-map
route-map Filter, permit, sequence 10
 Match clauses:
 Set clauses:
 local-preference 150
 Policy routing matches: 0 packets, 0 bytes
```

#### Step 2

On the ISP1 router, verify the BGP configuration.

```
ISP1# show running-config | section bgp
router bgp 100
 bgp log-neighbor-changes
 neighbor 172.16.11.1 remote-as 1
 neighbor 172.16.11.1 prefix-list Customer in
 neighbor 172.16.11.1 route-map Filter in
 neighbor 172.16.11.1 filter-list 10 in
 neighbor 172.16.12.2 remote-as 2
 neighbor 172.16.12.2 prefix-list Customer in
 neighbor 172.16.12.2 route-map Filter in
 neighbor 172.16.12.2 filter-list 10 in
 neighbor 172.16.13.3 remote-as 3
 neighbor 172.16.13.3 prefix-list Customer in
 neighbor 172.16.13.3 route-map Filter in
 neighbor 172.16.13.3 filter-list 10 in
```

All three neighbors are configured with the same parameters. Instead of configuring the same configuration three times, you can put all the configuration into the peer group configuration and assign all the neighbor to this peer group.

#### Step 3

On the ISP1 router, create a peer group CUSTOMERS.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor CUSTOMERS peer-group
```

#### Step 4

On the ISP1 router, move all per-neighbor BGP configuration into peer group configuration.

Remove the filter list 10, prefix list Customer, and route map Filter from all three neighbors. Apply this three routing policies to the CUSTOMERS peer group

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor CUSTOMERS prefix-list Customer in
ISP1(config-router)# neighbor CUSTOMERS route-map Filter in
ISP1(config-router)# neighbor CUSTOMERS filter-list 10 in
ISP1(config-router)# no neighbor 172.16.11.1 prefix-list Customer in
ISP1(config-router)# no neighbor 172.16.11.1 route-map Filter in
ISP1(config-router)# no neighbor 172.16.11.1 filter-list 10 in
ISP1(config-router)# no neighbor 172.16.12.2 prefix-list Customer in
ISP1(config-router)# no neighbor 172.16.12.2 route-map Filter in
ISP1(config-router)# no neighbor 172.16.12.2 filter-list 10 in
ISP1(config-router)# no neighbor 172.16.13.3 prefix-list Customer in
ISP1(config-router)# no neighbor 172.16.13.3 route-map Filter in
ISP1(config-router)# no neighbor 172.16.13.3 filter-list 10 in
```

Applying the routing policies only to the peer group simplifies the BGP configuration and improves BGP scalability. Instead of configuring the same parameters for each neighbor, you now configure them only once and then assign as many neighbor to the peer group as you want.

### Step 5

On the ISP1 router, assign all three neighbors to the CUSTOMERS peer group.

```
ISP1(config)# router bgp 100
ISP1(config-router)# neighbor 172.16.11.1 peer-group CUSTOMERS
ISP1(config-router)# neighbor 172.16.12.2 peer-group CUSTOMERS
ISP1(config-router)# neighbor 172.16.13.3 peer-group CUSTOMERS
```

This configuration enables the neighbors that are assigned to the peer group, to inherit common configuration from the peer group. You do not need to configure each individual neighbor.

## Monitoring Peer Groups

```
router# show ip bgp peer-group [peer-group-name]
```

- Displays the definition of the specified peer group or all peer groups

```
router# show ip bgp peer-group [peer-group-name] summary
```

- Displays summary status of all neighbors in the peer group

```
router# clear ip bgp [peer-group-name] [soft [in | out]]
```

- Clears BGP session with all peer group members

To display information about BGP peer groups, use the **show ip bgp peer-group** command. If you use **summary** keyword, only a summary of the status of all the members of a peer group will be shown.

To reset the BGP sessions with all the members of a peer group, use the **clear ip bgp** command.

```
clear ip bgp [* | neighbor-address | peer-group-name] [soft [in | out]]
```

### Syntax Description

| Parameter               | Description                                                      |
|-------------------------|------------------------------------------------------------------|
| *                       | Resets all current BGP sessions.                                 |
| <i>neighbor-address</i> | Resets only the specified BGP neighbor.                          |
| <i>peer-group-name</i>  | Resets only the specified BGP peer group.                        |
| <b>soft</b>             | (Optional) Initiates soft reconfiguration.                       |
| <b>in   out</b>         | (Optional) Triggers inbound or outbound soft reconfiguration. If |

| Parameter | Description                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------|
|           | you do not specify the in or out option, both inbound and outbound soft reconfiguration are triggered. |

## Step 6

On the ISP1 router, verify that all prefix list, filter list, and route map configuration is now applied to the peer group CUSTOMER.

```
ISP1# show ip bgp peer-group
BGP peer-group is CUSTOMERS
 BGP version 4
 Neighbor sessions:
 0 active, is not multisession capable (disabled)
 Default minimum time between advertisement runs is 30 seconds

For address family: IPv4 Unicast
 BGP neighbor is CUSTOMERS, peer-group external, members:
 172.16.11.1 172.16.12.2 172.16.13.3
 Index 0, Advertise bit 0
 Incoming update prefix filter list is Customer
 Incoming update AS path filter list is 10
 Route map for incoming advertisements is Filter
 Update messages formatted 0, replicated 0
 Number of NLRI in the update sent: max 0, min 0
```

You can see that all parameters configured for the peer group are listed in the output:

- Prefix list "Customer" for incoming updates.
- Filter list "10" for incoming updates.
- Route map "Filter" for incoming updates.

Note that the peer group should have only [IBGP](#) members or EBGp members. This peer group contains EBGp peers—172.16.11.1, 172.16.12.2, and 172.16.13.3. These are the peers that inherit the configuration from the peer group.

Another option to verify the configuration that is applied to the peer group is to simply view the BGP configuration on the ISP1.

```
ISP1# show running-config | section bgp
router bgp 100
 bgp log-neighbor-changes
 neighbor CUSTOMERS peer-group
 neighbor CUSTOMERS prefix-list Customer in
 neighbor CUSTOMERS route-map Filter in
 neighbor CUSTOMERS filter-list 10 in
 neighbor 172.16.11.1 remote-as 1
 neighbor 172.16.11.1 peer-group CUSTOMERS
 neighbor 172.16.12.2 remote-as 2
 neighbor 172.16.12.2 peer-group CUSTOMERS
 neighbor 172.16.13.3 remote-as 3
 neighbor 172.16.13.3 peer-group CUSTOMERS
```

## Step 7

On the ISP1 router, verify that R1, R2, and R3 peers are part of the CUSTOMERS peer group.

```
ISP1# show ip bgp peer-group CUSTOMERS summary
BGP router identifier 172.16.13.11, local AS number 100
BGP table version is 10, main routing table version 10
3 network entries using 444 bytes of memory
3 path entries using 192 bytes of memory
6/3 BGP path/bestpath attribute entries using 816 bytes of memory
3 BGP AS-PATH entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 1524 total bytes of memory
BGP activity 24/21 prefixes, 27/24 paths, scan interval 60 secs

Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ Up/Down State/PfxRcd
172.16.11.1 4 1 18 23 10 0 0 00:12:10 1
172.16.12.2 4 2 18 23 10 0 0 00:11:58 1
172.16.13.3 4 3 17 21 10 0 0 00:11:46 1
```

The output lists you all the members of CUSTOMERS peer group

- R1 (172.16.11.1)
- R2 (172.16.12.2)
- R3 (172.16.13.3)

Another option to verify that the neighbor is a member of the peer group is using the show ip bgp neighbor command.

```
ISP1# show ip bgp neighbor 172.16.11.1
BGP neighbor is 172.16.11.1, remote AS 1, external link
Member of peer-group CUSTOMERS for session parameters
 BGP version 4, remote router ID 10.1.0.1
 BGP state = Established, up for 00:12:39
 Last read 00:00:44, last write 00:00:28, hold time is 180, keepalive interval is 60 seconds
<... output omitted ...>
```

```
ISP1# show ip bgp neighbor 172.16.12.2
BGP neighbor is 172.16.12.2, remote AS 2, external link
Member of peer-group CUSTOMERS for session parameters
 BGP version 4, remote router ID 10.2.0.1
 BGP state = Established, up for 00:13:00
 Last read 00:00:13, last write 00:00:12, hold time is 180, keepalive interval is 60 seconds
<... output omitted ...>
```

```
ISP1# show ip bgp neighbor 172.16.13.3
BGP neighbor is 172.16.13.3, remote AS 3, external link
Member of peer-group CUSTOMERS for session parameters
 BGP version 4, remote router ID 10.3.0.1
 BGP state = Established, up for 00:13:08
 Last read 00:00:12, last write 00:00:28, hold time is 180, keepalive interval is 60 seconds
<... output omitted ...>
```

### Step 8

On the ISP1 router, verify that you still receive networks, that all three customers advertise. Verify that the local preference is still set to 150 for all the networks.

```
ISP1# show ip bgp
BGP table version is 10, local router ID is 172.16.13.11
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.1.0.0/24 172.16.11.1 0 150 0 1 i
*> 10.2.0.0/24 172.16.12.2 0 150 0 2 i
*> 10.3.0.0/24 172.16.13.3 0 150 0 3 i
```

You can see that ISP1 router in AS 100 received three networks, each from different customer:

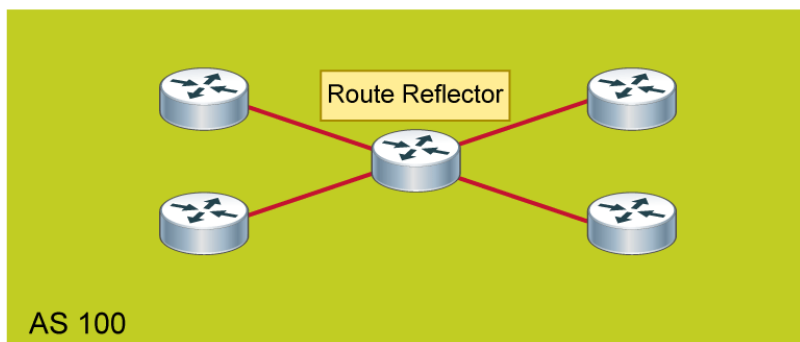
- 10.1.0.0/24 network originates in the first customer's AS 1.
- 10.2.0.0/24 network originates in the second customer's AS 2.
- 10.3.0.0/24 network originates in the third customer's AS 3.

All three networks still have local preference set to 150. Even though the route map that sets the local preference is now included in the peer group configuration, the neighbors inherited the configuration from the peer group.

## BGP Peer Group Configuration Examples

### Peer Group Configuration Example—BGP Route Reflector

In this example, a router acting as a [BGP](#) route reflector has four [IBGP](#) neighbors.



```
router(config)# router bgp 100
router(config-router)# neighbor IBGP_peers peer-group
router(config-router)# neighbor IBGP_peers remote-as 100
router(config-router)# neighbor IBGP_peers update-source loopback 0
router(config-router)# neighbor IBGP_peers password c73Dx8K
router(config-router)# neighbor IBGP_peers send-community
router(config-router)# neighbor 10.0.1.3 peer-group IBGP_peers
router(config-router)# neighbor 10.0.1.4 peer-group IBGP_peers
router(config-router)# neighbor 10.0.1.6 peer-group IBGP_peers
router(config-router)# neighbor 10.0.1.8 peer-group IBGP_peers
```

In a large [AS](#), some routers may have many IBGP sessions. A peer group that is named "IBGP\_peers" is created to handle all of the IBGP sessions. The peer group is created and configured with the remote AS, update-source, authentication, and community-passing parameters. When the actual neighbors are configured as members of the peer group, all these configuration parameters will apply to all of the neighbors.

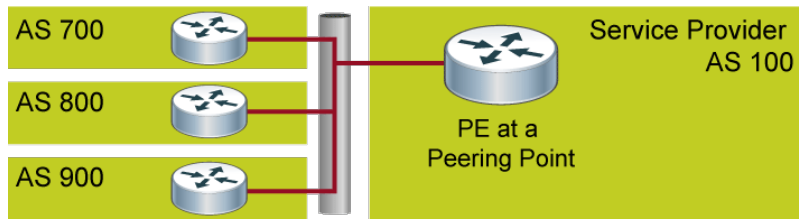
In the case of IBGP, the remote AS can also be part of the peer group configuration since the AS number is the same for all peer group members.

The peer group is a very powerful tool when you are dealing with many IBGP neighbors. You can give all of the neighbors the same configuration to ensure a consistent AS-wide routing policy.

### Peer Group Configuration Example—Edge Router at a Peering Point

In this example, the router in AS 100 is being configured with a peer group named "Peering"

```
PE(config)# router bgp 100
PE(config-router)# neighbor Peering peer-group
PE(config-router)# neighbor Peering filter-list 10 in
PE(config-router)# neighbor Peering route-map PeerMap out
PE(config-router)# neighbor Peering maximum-prefix 50
PE(config-router)# neighbor Peering remove-private-as
PE(config-router)# neighbor 209.165.207.7 remote-as 700
PE(config-router)# neighbor 209.165.207.7 peer-group Peering
PE(config-router)# neighbor 209.165.208.8 remote-as 800
PE(config-router)# neighbor 209.165.208.8 peer-group Peering
PE(config-router)# no neighbor 209.165.208.8 maximum-prefix 50
PE(config-router)# no neighbor 209.165.208.8 filter-list 10 in
PE(config-router)# neighbor 209.165.209.9 remote-as 900
PE(config-router)# neighbor 209.165.209.9 peer-group Peering
```



This peer group is used for all peer providers because they share an almost identical routing policy. The peer group is first created as a template, which is configured with an incoming AS path filter list (list 10) and an outgoing route map named "PeerMap." The maximum number of received prefixes is also set in the peer group to 50. The peer group has also been configured to remove private AS numbers from all AS paths before the routes are sent to the peer AS. These AS numbers are in the range 64512 to 65535 inclusive.

The neighbors in AS 700, AS 800, and AS 900 are then assigned to the peer group, meaning that the router in AS 100 will attempt to open BGP sessions with those routers. If the BGP sessions are successfully established, filter list 10 and the route map PeerMap, as configured in the peer group, will be applied to incoming and outgoing routes from both neighbors, respectively.

As defined in the router configuration, filter list 10 filters out any incoming routes from peer group members unless otherwise specified. However, in the case of the neighbor in AS 800, the individual configuration of **no filter-list 10** will override the peer group configuration. Thus, the filter list will not be used for this neighbor. The limitation on the number of received routes from AS 800 is also removed from the neighbor in AS 800.



The peer group is a very powerful tool when you are dealing with a large number of neighbors with almost identical configurations. However, if any of the customers that are assigned to the peer group require routing information that is different from other members of the peer group, then that neighbor must be removed from the peer group and configured individually.



## BGP Dynamic Update Peer Groups Feature

In the Cisco IOS 12.0(24)S release Cisco introduced dynamic update peer groups, a feature that enables the routers to automatically group BGP neighbor into groups and generate update messages accordingly.

- Separates BGP update generation from neighbor configuration
- Introduces a new algorithm that dynamically calculates and optimizes update groups of neighbors that share the same outbound policies.
- Requires no additional configuration.
- Optimal BGP update message generation occurs automatically and independently

In previous versions of Cisco IOS software, BGP update messages were grouped based on peer group configurations. This method of grouping neighbors for BGP update message generation reduced the amount of system processing resources that are needed to process the routing table. This method, however, had the following limitations:

- All neighbors that shared the same peer group configuration also had to share the same outbound routing policies.
- All neighbors had to belong to the same peer group and address family. Neighbors that are configured in different peer groups cannot belong to different address families.

These limitations existed to balance optimal update generation and replication against peer group configuration. These limitations also caused the network operator to configure smaller peer groups, which reduced the efficiency of update message generation.

The BGP dynamic update groups feature contains an algorithm that dynamically calculates and optimizes update groups of neighbors that share outbound policies and can share the update messages. The BGP dynamic update groups feature separates update group replication from peer group configuration, improving convergence time and flexibility of neighbor configuration. The BGP update groups require no configuration, and a router optimizes BGP update message generation automatically.

When a change to the configuration occurs, the router automatically recalculates update group memberships and applies the changes by triggering an outbound soft reset after a 3-minute timer expires. This behavior is designed to provide the network operator with time to change the configuration if a mistake is made.



A soft reset, which is performed on a per-neighbor basis, does not clear the BGP session and facilitates the application of new policies. There are two methods of performing a soft reset. Dynamic inbound soft reset is used to generate inbound updates from a neighbor. An outbound soft reset is used to send a new set of updates to a neighbor.

```
clear ip bgp update-group [index-group | ip-address]
```

- Clears BGP update-group member sessions.

```
debug ip bgp groups [index-group | ip-address]
```

- Displays information that is related to the processing of BGP update-groups.

```
show ip bgp replication [index-group | ip-address]
```

- Displays update replication statistics for BGP update-groups.

```
show ip bgp update-group [index-group | ip-address] [summary]
```

- Displays information about BGP update-groups.

If no argument is specified, the **clear ip bgp update-group** command recalculates all update-groups. Specific index numbers for update groups and information about

update-group membership is displayed in the output of the **show ip bgp update-group** and **debug ip bgp groups** commands.

## clear ip bgp update-group

To clear BGP update-group member sessions, use the **clear ip bgp update-group** command in privileged EXEC mode.

```
clear ip bgp update-group [index-group | ip-address]
```

### Syntax Description

| Parameter          | Description                                                                                                                                                |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>index-group</i> | (Optional) Specifies that the update-group with corresponding index number will be reset. The range of update-group index numbers is from 1 to 4294967295. |
| <i>ip-address</i>  | (Optional) Specifies the <a href="#">IP address</a> of a single peer that will be reset.                                                                   |

## debug ip bgp groups

The output of the **debug ip bgp groups** command displays you information about update-group calculations and the addition and removal of update-group members. Information about peer groups and peer-policy and peer-session templates is also displayed in the output of this command as neighbor configurations change.

```
debug ip bgp groups [index-group | ip-address]
```

### Syntax Description

| Parameter          | Description                                                                                                                                                                         |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>index-group</i> | (Optional) Specifies that update-group debugging information for the corresponding index number will be displayed. The range of update-group index numbers is from 1 to 4294967295. |
| <i>ip-address</i>  | (Optional) Specifies that update-group debugging information for a single peer will be displayed.                                                                                   |

The following example output from the **debug ip bgp groups** command shows the recalculation of update-groups after the **clear ip bgp groups** command was issued:

```
ISP1# debug ip bgp groups
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.11.1 Down User reset
5w4d: BGP-
DYN(0): Comparing neighbor 172.16.11.1 flags 0x0 cap 0x0 and updgrp 2 f10
5w4d: BGP-DYN(0): Update-
group 2 flags 0x0 cap 0x0 policies same as 172.16.11.1 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Down User reset
5w4d: BGP-
DYN(0): Comparing neighbor 172.16.12.2 flags 0x0 cap 0x0 and updgrp 2 f10
5w4d: BGP-DYN(0): Update-
group 2 flags 0x0 cap 0x0 policies same as 172.16.12.2 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.100.22 Down User reset
5w4d: BGP-
DYN(0): Comparing neighbor 172.16.100.22 flags 0x0 cap 0x0 and updgrp 2 f1
0
5w4d: BGP-DYN(0): Update-
group 2 flags 0x0 cap 0x0 policies same as 172.16.100.22 f10
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.11.1 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.100.22 Up
5w4d: %BGP-5-ADJCHANGE: neighbor 172.16.12.2 Up
```



The output of this command can be very verbose, so you should not deploy this command in a production network unless you are troubleshooting a problem.

## show ip bgp replication

To display update replication statistics for BGP update-groups, use the **show ip bgp replication** command.

```
show ip bgp replication [index-group | ip-address]
```

## Syntax Description

| Parameter          | Description                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>index-group</i> | (Optional) Specifies that update replication statistics for the update-group with corresponding index number will be displayed. The range of update-group index numbers is from 1 to 4294967295. |
| <i>ip-address</i>  | (Optional) Specifies the IP address of a single neighbor for which update-group statistics will be displayed.                                                                                    |

The following sample output from the **show ip bgp replication** command shows update-group replication information for all for neighbors:

|                               |         |               |        |         |         |        |
|-------------------------------|---------|---------------|--------|---------|---------|--------|
| ISP1# show ip bgp replication |         |               |        |         |         |        |
|                               |         |               |        |         | Current |        |
| t                             | Next    |               |        |         |         |        |
| Index                         | Members | Leader        | MsgFmt | MsgRepl | Csize   | Versio |
| n Version                     |         |               |        |         |         |        |
| 1                             | 2       | 172.16.11.1   | 0      | 0       | 0/1000  | 1      |
| /0                            |         |               |        |         |         |        |
| 2                             | 1       | 172.16.100.22 | 0      | 0       | 0/1000  |        |
| 1/0                           |         |               |        |         |         |        |

## show ip bgp update-group

To display information about BGP update-groups, use the **show ip bgp update-group**.

```
show ip bgp update-group [index-group | ip-address] [summary]
```

## Syntax Description

| Parameter          | Description                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>index-group</i> | (Optional) Specifies that update replication statistics for the update-group with corresponding index number will be displayed. The range of update-group index numbers is from 1 to 4294967295.                |
| <i>ip-address</i>  | (Optional) Specifies the IP address of a single neighbor for which update-group statistics will be displayed.                                                                                                   |
| <b>summary</b>     | (Optional) Displays a summary of update-group member information.<br>The output can be filtered to show information for a single index-group or peer with the <i>index-group</i> or <i>ip-address</i> argument. |

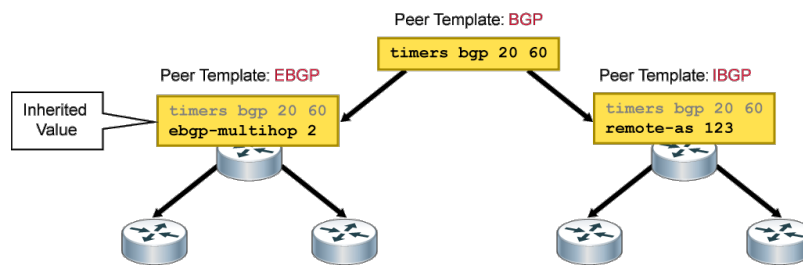
The following sample output from the **show ip bgp update-group** command shows update-group information for all neighbors:

|                                                                             |  |
|-----------------------------------------------------------------------------|--|
| ISP1# show ip bgp update-group                                              |  |
| BGP version 4 update-group 1, external, Address Family: IPv4 Unicast        |  |
| BGP Update version : 6/0, messages 0                                        |  |
| Topology: global, highest version: 6, tail marker: 6                        |  |
| Format state: Current working (OK, last minimum advertisement interval)     |  |
| Refresh blocked (not in list, last not in list)                             |  |
| Update messages formatted 1, replicated 2, current 0, refresh 0, limit 1000 |  |
| Number of NLRI's in the update sent: max 5, min 0                           |  |
| Minimum time between advertisement runs is 30 seconds                       |  |
| Has 1 member:                                                               |  |
| 172.16.100.22                                                               |  |
| BGP version 4 update-group 2, external, Address Family: IPv4 Unicast        |  |
| BGP Update version : 6/0, messages 0                                        |  |
| Topology: global, highest version: 6, tail marker: 6                        |  |
| Format state: Current working (OK, last minimum advertisement interval)     |  |
| Refresh blocked (not in list, last not in list)                             |  |
| Route map for outgoing advertisements is CUST1                              |  |
| Update messages formatted 1, replicated 2, current 0, refresh 0, limit 1000 |  |
| Number of NLRI's in the update sent: max 5, min 0                           |  |
| Minimum time between advertisement runs is 30 seconds                       |  |
| Has 2 members:                                                              |  |
| 172.16.12.2 172.16.11.1                                                     |  |



## BGP Peer Templates Overview

The BGP dynamic update groups feature separates peer group configuration from update group generation. BGP neighbor configuration is no longer restricted by outbound routing policies, and update groups can belong to different address families.



- Peer templates contain configuration pattern that can be applied to neighbors that share common policies.
- They are reusable and support inheritance, which allows you to group and apply distinct neighbor configurations for BGP neighbors.
- You can define very complex configuration patterns through the use of inheritance.

Even though BGP update message generation has been separated from peer group configuration, peer group configuration still has the following limitations:

- A neighbor can belong only to one peer group.
- Neighbors that belong to different address families cannot belong to the same peer group.
- Different sets of policies cannot be grouped and applied to a neighbor.

To address the limitations of peer groups, the BGP peer templates feature was introduced along with the BGP dynamic update groups feature.

A peer template is a configuration pattern that can be applied to neighbors that share common policies. Peer templates are reusable and support inheritance, which allows you to group and apply distinct neighbor configurations for BGP neighbors that share common policies. Peer templates also allow you to define very complex configuration patterns, since a peer template can inherit a configuration from another peer template.

The example in the figure presents peer template inheritance. One peer template, called BGP, is configured and contains parameters that are common to all BGP neighbors. In the example, such parameters are the BGP keepalive and hold-down timers. Then one peer template is configured for IBGP and one for EBG' neighbors. Both templates inherit settings from the common peer template that is called BGP and add other parameters that are common only to IBGP or EBG' settings, respectively. Therefore templates can be created more effectively with fewer configurations when compared to BGP peer groups.

Two types of peer templates are available:

- **Peer session template:** Used to group and apply the configuration of general session commands that are common to all address-family configuration modes.
- **Peer policy template:** Used to group and apply the configuration of commands that are applied within specific address-family configuration modes.

There are two types of peer templates:

- Peer session templates are used to group and apply the configuration of general session commands that are common to all address-families configuration modes.
- Peer policy templates are used to group and apply the configuration of commands that are applied within specific address-families configuration modes.

Peer templates improve the flexibility and enhance the capability of neighbor configuration. Peer templates also provide an alternative to peer group configuration and overcome some limitations of peer groups. BGP peer routers using peer templates also benefit from automatic update group configuration. With the configuration of the BGP peer templates the support of the BGP dynamic update

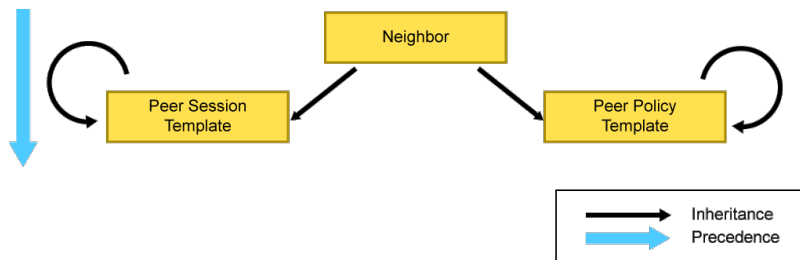
peer groups feature, you no longer need to configure peer groups in BGP. You benefit from improved configuration flexibility and faster convergence.



The configuration of BGP peer templates does not conflict with or restrict peer group configuration, and peer groups are still supported in Cisco IOS Releases that support BGP peer templates. However, a BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit policies from peer templates.

## BGP Peer Templates Inheritance

The inheritance capability is a key component of peer-template operation. Inheritance in a peer template is similar to the node and tree structures that are commonly found in general computing—for example, file and directory trees. A peer template can directly or indirectly inherit a configuration from another peer template. The directly inherited peer template represents the tree in the structure, and the indirectly inherited peer template represents a node in the tree. Because each node also supports inheritance, branches can be created that apply the configurations of all indirectly inherited peer templates within a chain that traces back to the directly inherited peer template or the source of the tree. This structure eliminates the need to repeat configuration statements that are commonly reapplied to groups of neighbors. Common configuration statements can now be applied once and then indirectly inherited by peer templates that are applied to neighbor groups with common configurations.



The BGP peer templates inheritance characteristics are as follows:

- A session template can inherit configuration from another session template.
- A policy template can inherit configuration from another policy template.
- Neighbors can inherit from a session and a policy template.

Inheritance expands the scalability and flexibility of neighbor configuration. It allows you to chain together peer-template configurations to create simple configurations that inherit common configuration statements or complex configurations that apply specific configuration statements along with common inherited configurations.

General session commands can be configured once in a peer session template and then applied to many neighbors through the direct application of a peer session template. Or through indirect inheritance from a peer session template. The configuration of peer session templates simplifies the configuration of general session commands that are commonly applied to all neighbors within an [AS](#).

Peer session templates support direct and indirect inheritance. A peer can be configured with only one peer session template at a time, and that peer session template can contain only one indirectly inherited peer session template.



If you attempt to configure more than one inherit statement with a single peer session template, an error message will be displayed.

This behavior allows a [BGP](#) neighbor to directly inherit only one session template and indirectly inherit up to seven additional peer session templates. This practice allows you to apply a maximum of eight peer session configurations to a neighbor. The configuration from the directly inherited peer session template, and the configurations from up to seven indirectly inherited peer session templates. Inherited peer session configurations are evaluated first and applied starting with the last node in the branch and ending with the directly applied peer session template configuration at the source of the tree. The directly applied peer session template will have priority over inherited peer session template configurations. Any configuration statements that are duplicated in inherited peer session templates will be overwritten by the directly applied peer session template. Meaning that if a general session command is reapplied with a different value, the subsequent value will have priority and overwrite the previous value that was configured in the indirectly inherited template.

Peer policy templates are used to configure BGP policy commands that are configured for neighbors that belong to specific address families. Like peer session templates, peer policy templates are configured once and then applied to many neighbors through the direct application of a peer policy template or through

inheritance from peer policy templates. The configuration of peer policy templates simplifies the configuration of BGP policy commands that are applied to all neighbors within an AS

Like peer session templates, a peer policy template supports inheritance. However, there are minor differences. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. That is, a total of eight peer policy templates can be applied to a neighbor or neighbor group. Inherited peer policy templates are configured with sequence numbers like route maps. An inherited peer policy template, like a route map, is evaluated starting with the inherit statement with the lowest sequence number and ending with the highest sequence number. However, the difference is that a peer policy template will not collapse like a route map. Every sequence is evaluated, and if a BGP policy command is reapplied with a different value, any previous value from a lower sequence number will be overwritten.

The directly applied peer policy template and the inherit statement with the highest sequence number will always have priority and be applied last. Commands that are reapplied in subsequent peer templates will always overwrite the previous values. This behavior is designed to allow you to apply common policy configurations to large neighbor groups and specific policy configurations only to certain neighbors and neighbor groups, without duplicating individual policy configuration commands.

The configuration of peer policy templates simplifies and improves the flexibility of BGP configuration. A specific policy can be configured once and referenced many times. Because a peer policy supports up to eight levels of inheritance, specific and complex BGP policies can also be created.



## BGP Peer Templates Configuration

General session commands that are common for neighbors that are configured in different address families can be configured within the same peer session template. Peer session templates are created and configured in peer session configuration mode. Only general session commands can be configured in a peer session template. [BGP](#) policy configuration commands that are configured only for specific address families configuration modes are configured with peer policy templates.

### Peer Session Commands

```
template peer-session session-template-name
```

- Creates a peer session.

```
inherit peer-session session-template-name
```

- Configures a peer session template to inherit the configuration of another peer session template.

```
neighbor ip-address inherit peer-session session-template-name
```

- Sends a peer session template to a neighbor so that the neighbor can inherit the configuration.

To create a peer session template and enter session template configuration mode, use the **template peer-session** command in router configuration mode.

```
template peer-session session-template-name
```

#### Syntax Description

| Parameter                    | Description                               |
|------------------------------|-------------------------------------------|
| <i>session-template-name</i> | Name or tag for the peer session template |

To configure a peer session template to inherit the configuration of another peer session template, use the **inherit peer-session** command in session-template configuration mode.

```
inherit peer-session session-template-name
```

#### Syntax Description

| Parameter                    | Description                               |
|------------------------------|-------------------------------------------|
| <i>session-template-name</i> | Name or tag for the peer session template |

The **neighbor inherit peer-session** command is used to send locally configured session templates to the specified neighbor. If the session template is configured to inherit configurations from other session templates, the specified neighbor will also indirectly inherit these configurations from the other session templates. A neighbor can directly inherit only one peer session template and indirectly inherit up to seven peer session templates.

Use the **neighbor inherit peer-session** command in address family or router configuration mode.

```
neighbor ip-address inherit peer-session session-template-name
```

#### Syntax Description

| Parameter                    | Description                                |
|------------------------------|--------------------------------------------|
| <i>ip-address</i>            | <a href="#">IP address</a> of the neighbor |
| <i>session-template-name</i> | Name or tag for the peer session template  |



A BGP neighbor cannot be configured to work with both peer groups and peer templates. A BGP neighbor can be configured to belong only to a peer group or to inherit

policies only from peer templates.

## Peer Policy Commands

Peer policy templates support only general policy commands. BGP policy configuration commands that are configured only for specific address families configuration modes are configured with peer policy templates.

```
inherit peer-policy policy-template-name sequence-number
```

- Configures a peer policy template to inherit the configuration from another peer policy template.

```
neighbor ip-address inherit peer-policy policy-template-name
```

- Sends a peer policy template to a neighbor so that the neighbor can inherit the configuration.

To configure a peer policy template to inherit the configuration from another peer policy template, use the **inherit peer-policy** command in policy-template configuration mode.

```
inherit peer-policy policy-template-name sequence-number
```

### Syntax Description

| Parameter               | Description                                                                                                                                                          |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>peer-policy-name</i> | Name of the peer policy template to be inherited.                                                                                                                    |
| <i>sequence-number</i>  | Sequence number that sets the order in which the peer policy template is evaluated. Like a route map sequence number, the lowest sequence number is evaluated first. |

The **neighbor inherit peer-policy** command is used to send locally configured policy templates to the specified neighbor. If the policy template is configured to inherit configurations from other peer policy templates, the specified neighbor will also indirectly inherit these configurations from the other peer policy templates. A directly applied peer policy template can directly or indirectly inherit configurations from up to seven peer policy templates. So, a total of eight peer policy templates can be applied to a neighbor or neighbor group.

```
neighbor ip-address inherit peer-policy policy-template-name
```

### Syntax Description

| Parameter                   | Description                              |
|-----------------------------|------------------------------------------|
| <i>ip-address</i>           | IP address of the neighbor               |
| <i>policy-template-name</i> | Name or tag for the peer policy template |

## BGP Peer Templates Verification

```
show ip bgp template peer-session [session-template-name]
```

- Displays peer policy template configurations.

```
show ip bgp template peer-policy [policy-template-name]
```

- Displays locally configured peer policy templates.

The **show ip bgp template peer-session** command is used to display locally configured peer session templates. The output can be filtered to display a single peer session template with the peer-session-name argument. This command also supports all standard output modifiers.

```
show ip bgp template peer-session [session-template-name]
```

### Syntax Description

| Parameter                    | Description                                                   |
|------------------------------|---------------------------------------------------------------|
| <i>session-template-name</i> | (Optional) Name of a locally configured peer session template |

To display locally configured peer policy templates, use the **show ip bgp template peer-policy** command in EXEC mode.

```
show ip bgp template peer-policy [policy-template-name]
```

#### Syntax Description

| Parameter                   | Description                                                  |
|-----------------------------|--------------------------------------------------------------|
| <i>policy-template-name</i> | (Optional) Name of a locally configured peer policy template |

## Summary

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

- BGP peer groups were designed primarily for CPU optimization. They can also be used for configuration optimization.
  - BGP peer groups are used on a router with several neighbors that have similar, but not completely identical policies.
- Peer groups have limitations because of the way that they are used to build BGP updates:
  - Per-neighbor BGP parameters that affect outbound updates cannot be changed for peer group members.
  - IBGP and EBGP neighbors cannot be mixed in a peer group.
- To configure BGP peer groups, create a BGP peer group, specify parameters for the BGP peer group, create a BGP neighbor, and then assign a neighbor to the peer group.
- BGP dynamic update groups separate BGP update generation from neighbor configuration.
- BGP peer templates are reusable configuration patterns that support inheritance.
  - Peer session templates—contain configuration that is common to all address families.
  - Peer policy templates—contain configuration that is applied within a specific address family.

# Using BGP Route Dampening

## Overview

Even when a [BGP](#) implementation is correctly configured and highly robust, the performance of the routing process on any given router is limited. Limiting the propagation of unstable routes, specifically when they are not beneficial to the network, becomes an important issue because it reduces the processing requirements of the router that is forced to process routing table state changes.

Route dampening is a BGP feature that has been designed to reduce BGP processing requirements by minimizing the propagation of unstable routes to BGP peers. AS border routers, in any BGP implementation, cannot rely upon external peers to sufficiently shield the AS from routing table instability. Route dampening allows route instability to be contained at an [AS](#) border router that borders the instability.

This lesson describes the purpose and operation of the route-dampening feature on Cisco IOS routers. Also discussed in this lesson are the Cisco IOS commands that are required to enable route dampening, modify default dampening parameters, and release a route that has been suppressed because of dampening. The Cisco IOS commands that are used to monitor route dampening are also discussed.

Upon completing this lesson, you will be able to:

- Describe the purpose of BGP route dampening
- Describe the operation of BGP route dampening
- Identify the Cisco IOS commands that are required to configure BGP route dampening
- Identify the Cisco IOS commands that are required to release dampened routes and monitor BGP route-dampening

## BGP Route Dampening

BGP is the only routing protocol that is designed for large internetworks with the specific intention of carrying a large number of prefixes. There are several mechanisms that are built into BGP that ensure maximum router stability.

- Designed to reduce router processing load that is caused by unstable routes
- Prevents sustained routing oscillations without affecting other well-behaved routes
- Defined in RFC 2439: *BGP Route Flap Dampening*
- A tool that is designed to help minimize the number of BGP updates
- Other update reduction tools:
  - Batching of BGP updates
  - Per-neighbor update timers

For example, a BGP router does not forward BGP routing updates immediately after receiving them. Every time a BGP router sends an update, it starts a 5-second timer for internal neighbors and a 30-second timer for external neighbors. No new updates can be sent until that timer expires. The result is that if a router contains a link that is flapping repeatedly (available, then unavailable, then available, then unavailable, and so on) at a rate of once per second, external routers see the flap at a much slower rate. Routers that are external to the source of the flap are not forced to recalculate the best path every second but, at most, every 30 seconds.

Reducing the rate at which neighboring routers process flapping routes helps with reducing the requirements to process the BGP update. However, routers that process routing updates for unstable routes are still wasting resources in determining the best route to the destination. Because the unstable route is oscillating between up and down, each route update that a router receives causes it to process the unstable route all over again. A better approach is to remove the update about the route until it can be guaranteed that the destination is more stable.

With this goal in mind, an extra BGP scalability mechanism, route flap dampening, was created to reduce route update processing requirements by suppressing unstable routes.

- Minimizes the amount of BGP update processing in the Internet by suppressing unstable (flapping) routes
- Does not suppress routes that occasionally flap
- Suppresses routes that are likely to flap in the future based on the history of their behavior
  - Flap = Remove route
  - Suppress = Do not use a route after it reappears

Most service providers hold routing information for the entire Internet. Therefore, a flapping link somewhere in the Internet can cause all routers in the Internet to keep processing changes because of one single link. If, however, one of the autonomous systems in the Internet implements route dampening, the flapping network is suppressed. The route is not propagated further to other autonomous systems until the configured rules of route dampening allow it.

A "flap" refers to a route that is repeatedly available, then unavailable, then available, then unavailable, and so on. If a route flaps once or twice, it is typically not considered a flap from an administrative perspective. If the flapping happens more often, however, there is probably something wrong with the destination and the route should be suppressed. The BGP router stores a suppressed route in the BGP table but does not consider it in the BGP path-selection process and does not therefore propagate it to other BGP neighbors or use it for data forwarding.

## BGP Route Dampening Operation

A BGP router with route dampening enabled keeps track of all routes (even the routes that are unreachable) so it can recall the penalties that are assigned to each route.

- Each time an EBGp route flaps, it gets 1000 penalty points (IGBP routes are not dampened).
- The penalty that is placed on a route decays according to the exponential decay algorithm.
- When the penalty exceeds the suppress limit, the route is dampened (no longer used or propagated to other neighbors).
- A dampened route is propagated when the penalty drops below the reuse limit.

Every time a route flap occurs, the flapping route receives 1000 penalty points. The penalty is gradually decreased by using a decaying algorithm. If a route flaps several times, it will be penalized (gain enough penalty points) and subsequently reach and exceed the suppress limit.

Any route that reaches the suppress limit is no longer forwarded to other neighbors until the assigned penalty is once again below the reuse limit. An exponential decay algorithm reduces penalty points that are applied to a flapping route. After the number of penalty points that are assigned to a route falls below the reuse limit, the BGP router once again advertises the route.

- The flap history is forgotten, when the penalty drops below half of the reuse limit.
- A route is never dampened for more time than the maximum suppress limit.
- An unreachable route with a flap history is put in the history state—it stays in the BGP table but only to maintain the flap history.
- A penalty is applied on the individual path in the BGP table, not on the IP prefix.

A router stops tracking penalty points when they are below half of the reuse limit.

The maximum suppress limit defines the maximum duration that a route can be suppressed after it has been suppressed.

After route dampening is enabled, the router never removes a route from the BGP table. A route that a BGP neighbor has withdrawn can still be seen in the BGP table and is marked with an "h" (history state).

A penalty is always applied to a path and not a prefix. If one of the paths is flapping, it does not mean that the destination is flapping.

## Configuring BGP Route Dampening

This topic identifies the Cisco IOS commands that are required to configure [BGP](#) route dampening.

```
router(config-router)# bgp dampening [half-life reuse suppress max-suppress-time] [route-map map-name]
```

- Configures BGP route dampening
- BGP dampening parameters:
  - *half-life* Decay time in which the penalty is halved
  - *suppress* Value when the route starts dampening
  - *reuse* Value when the dampened route is reused
  - *max-suppress-time* Maximum time to suppress the route
  - **route-map** Name of route-map controlling dampening

To enable route dampening, use the **bgp dampening** command. Optionally, you can change the default settings of the route-dampening parameters.

Route flap dampening requires the following parameters:

- *half-life*: The half-life is the time that is needed for the penalty to halve (default is 15 minutes).
- *suppress*: When a route has more penalty points than the suppress limit, the route is suppressed (default is 2000).
- *reuse*: After the flapping has stopped and the penalty for a route has fallen below the reuse limit, the route is unsuppressed (default is 750).
- *max-suppress-time*: No route can be suppressed longer than the max-suppress-time minutes (default is 1 hour; maximum is 255 minutes).

You can specify the four route flap-dampening parameters directly with the **bgp dampening** command. Alternatively, you can create a route-map that specifies different dampening parameters for different sets of routes, and then you can apply the route-map with the **bgp dampening route-map** command.

Most Internet service providers use default values:

- A flapping route is dampened after three successive flaps.
- A route stays suppressed for approximately 30 minutes.
- **Net result**: The route is lost for 30 minutes if a BGP session with a neighbor is cleared three times in succession.
- Default dampening parameter values are:
  - *half-life* 15 minutes
  - *suppress* 2000
  - *reuse* 750
  - *max-suppress-time* 60 minutes (4x half-life)
  - *per-flap penalty* 1000 (nonconfigurable)

This sample calculation shows how long a route that flaps three times is suppressed with the default values of the Cisco IOS route-dampening parameters. Each time that a route flaps, it accumulates another 1000 points. After the third flap, the route has almost 3000 points. Remember that the penalty is gradually decreased by using a decaying algorithm, causing a reduction in the number of points that the route accumulates. It takes 15 minutes for the penalty to drop below 1500 (provided there are no further flaps) and another 15 minutes to drop below the reuse limit of 750.

Many service providers change the default parameters to allow a maximum suppress time of several hours.



Using the **clear ip bgp \*** command is regarded as a flap to neighboring autonomous systems. Using this command several times may cause neighboring autonomous systems to suppress prefixes for some time even if there is nothing wrong with the route.



Using the **clear ip bgp \* [soft] [in | out]** command is not regarded as a flap, and this command should be used instead of **clear ip bgp \*** for clearing the neighbor



relationships.

You can change all default values if you specify them in the optional parameters of the **bgp dampening** command. The per-flap penalty is the only value that is not configurable.

```
router(config-route-map)# set dampening half-life reuse suppress max-suppress-time
```

- This command sets the BGP dampening parameters for individual routes that a route map entry matches.
- Apply this route-map to the **bgp dampening** command instead of specifying individual parameters.
- Applications:
  - Less aggressive dampening of routes toward root DNS servers (or other servers)
  - Dampening of smaller prefixes more aggressively
  - Selective dampening based on BGP neighbor and route-map match criteria

Many service providers prefer to implement selective dampening. Larger prefixes are usually less likely to flap and should not be penalized as aggressively as the smaller prefixes that populate most of the BGP table.

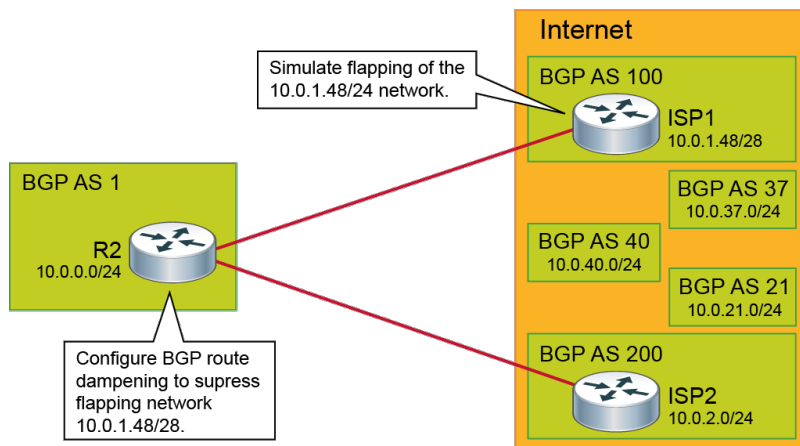
You can use a route-map in combination with a prefix-list to match on prefix length and to set different route-dampening parameters for larger prefixes than for smaller ones. A practical service provider policy is to use a route-map to exclude root [DNS](#) servers from dampening altogether.

You can then attach the route-map to the BGP route-dampening process with the **bgp dampening route-map** command.

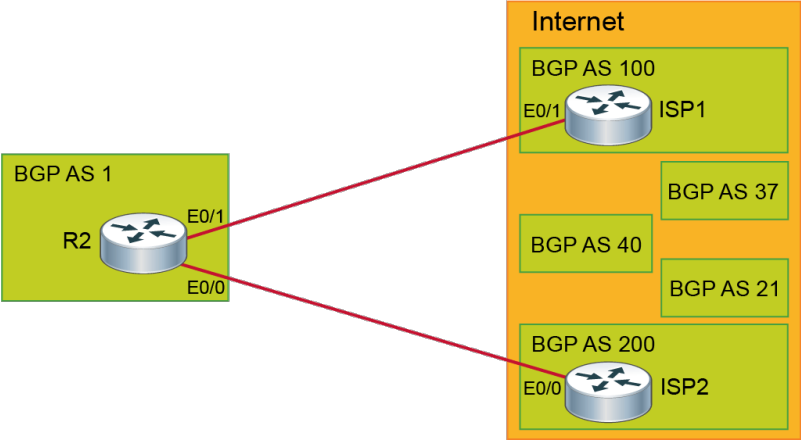
## Discovery 20: Configure BGP Route Dampening

### Overview

Through this discovery, you will learn how to configure [BGP](#) route dampening. You will enable route dampening on the R2 router. You will simulate flapping of one of the advertised networks on the ISP1 router. Consequently, the network flapping will get suppressed on the R2 router, where route dampening is enabled.



Topology



## Job Aids

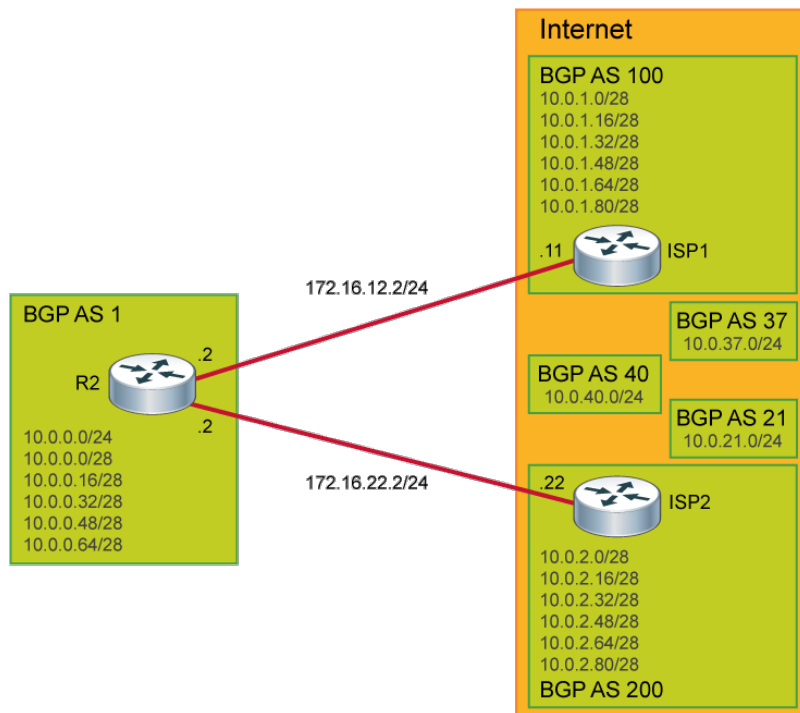


If you shut down an interface on a real router or switch, the connected device will see it as "down/down." Due to virtualization specifics, [IOL](#) behavior is slightly different. If you shut down an interface on a router or switch, the connected device will see it as "up/up." In IOL, the status of an interface can only be "up/up" or "administratively down/down."

### Device Information

| Device | Interface                                                                        | IP address                                                                                  | Description                                                        |
|--------|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| R2     | Ethernet 0/0                                                                     | 172.16.22.2/24                                                                              | Connection to ISP2                                                 |
| R2     | Ethernet 0/1                                                                     | 172.16.12.2/24                                                                              | Connection to ISP1                                                 |
| R2     | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4<br>Loopback 5               | 10.0.0.1/28<br>10.0.0.17/28<br>10.0.0.33/28<br>10.0.0.49/28<br>10.0.0.65/28                 | Loopbacks simulate <a href="#">LAN</a> networks                    |
| ISP1   | Ethernet 0/1                                                                     | 172.16.12.11/24                                                                             | Connection to R2                                                   |
| ISP1   | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4<br>Loopback 5<br>Loopback 6 | 10.0.1.1/28<br>10.0.1.17/28<br>10.0.1.33/28<br>10.0.1.49/28<br>10.0.1.65/28<br>10.0.1.81/28 | Loopbacks simulate LAN networks                                    |
| ISP1   | Loopback 21<br>Loopback 37<br>Loopback 40                                        | 10.0.21.1<br>10.0.37.1<br>10.0.40.1                                                         | Loopbacks simulate extra networks in different autonomous systems. |
| ISP2   | Ethernet 0/0                                                                     | 172.16.22.22/24                                                                             | Connection to R2                                                   |
| ISP2   | Loopback 1<br>Loopback 2<br>Loopback 3<br>Loopback 4<br>Loopback 5<br>Loopback 6 | 10.0.2.1/28<br>10.0.2.17/28<br>10.0.2.33/28<br>10.0.2.49/28<br>10.0.2.65/28<br>10.0.2.81/28 | Loopbacks simulate LAN networks                                    |
| ISP2   | Loopback 21<br>Loopback 37<br>Loopback 40                                        | 10.0.21.1<br>10.0.37.1<br>10.0.40.1                                                         | Loopbacks simulate extra networks in different ASs.                |

[IP addresses](#) and advertised networks in BGP are preconfigured as shown below:



BGP is also preconfigured as [EBGP](#) (R2 to ISP1 and R2 to ISP2). The ISP1 router announces these networks:

- 10.0.1.0/28
- 10.0.1.16/28
- 10.0.1.32/28
- 10.0.1.48/28
- 10.0.1.64/28
- 10.0.1.80/28

# Configure BGP Route Dampening

## Discovery Steps

### Step 1

On the R2 router, verify that you receive 10.0.1.48/28 route to the BGP table from the ISP1 neighbor.

```
R2# show ip bgp
BGP table version is 24, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/24 0.0.0.0 0 32768 i
*> 10.0.0.16/28 0.0.0.0 0 32768 i
*> 10.0.0.32/28 0.0.0.0 0 32768 i
*> 10.0.0.48/28 0.0.0.0 0 32768 i
*> 10.0.0.64/28 0.0.0.0 0 32768 i
*> 10.0.1.0/28 172.16.12.11 0 0 100 i
*> 10.0.1.16/28 172.16.12.11 0 0 100 i
*> 10.0.1.32/28 172.16.12.11 0 0 100 i
*> 10.0.1.48/28 172.16.12.11 0 0 100 i
*> 10.0.1.64/28 172.16.12.11 0 0 100 i
*> 10.0.1.80/28 172.16.12.11 0 0 100 i
<... output omitted...>
```

### Step 2

Enable BGP route dampening on the R2 router with default values for dampening parameters.

```
R2(config)# router bgp1
R2(config-router)# bgp dampening
```

### Step 3

Enable debugging for BGP dampening on the R2 router.

```
R2# debug ip bgp dampening
```

### Step 4

On the ISP1 router, disable and enable interface Loopback 4 (10.0.1.49/28) for at least three times. Allow yourself at least a minute before next command is applied.

```
ISP1(config)# interface Loopback 4
ISP1(config-if)# shutdown
!wait for at least 30 seconds!
ISP1(config-if)# no shutdown
!wait for 30 seconds!
!repeat these steps for at least three times!
```

**NOTE:** You will simulate the flapping of the 10.0.1.48/28 network by enabling and disabling loopback interface few times in short amount of time.

### Step 5

Observe the debugging output. Each time that you disable the interface Loopback 4, the 10.0.1/48 network with existing path gets charged with 1000 points. Once the suppress penalty passes the threshold of 2000 and you enable the interface Loopback 4, network 10.0.1.0/48 gets suppressed.

```
R2#
*Apr 1 08:36:29.423: EvD: charge penalty 1000, new accum. penalty 1000, flap count 1
*Apr 1 08:36:29.423: EvD: unsuppress item left in reuse timer array with penalty 1000
*Apr 1 08:36:29.423: BGP(0): charge penalty for 10.0.1.48/28 path 100 with half-life-
time 15 reuse/suppress 750/2000
*Apr 1 08:36:29.423: BGP(0): flapped 1 times since 00:00:00. New penalty is 1000
```

```

R2#
*Apr 1 08:37:12.319: EvD: accum. penalty decayed to 969 after 42 second(s)
R2#
*Apr 1 08:37:30.196: EvD: accum. penalty 957, not suppressed
R2#
*Apr 1 08:38:01.030: EvD: accum. penalty decayed to 935 after 31 second(s)
*Apr 1 08:38:01.030: EvD: charge penalty 1000, new accum. penalty 1935, flap count 2
*Apr 1 08:38:01.030: EvD: unsuppress item left in reuse timer array with penalty 1935
*Apr 1 08:38:01.030: BGP(0): charge penalty for 10.0.1.48/28 path 100 with half-life-
time 15 reuse/suppress 750/2000
*Apr 1 08:38:01.030: BGP(0): flapped 2 times since 00:01:31. New penalty is 1935
R2#
*Apr 1 08:38:12.380: EvD: accum. penalty decayed to 1920 after 11 second(s)
R2#
*Apr 1 08:38:31.858: EvD: accum. penalty 1890, not suppressed
R2#
*Apr 1 08:39:02.687: EvD: accum. penalty decayed to 1846 after 31 second(s)
*Apr 1 08:39:02.687: EvD: charge penalty 1000, new accum. penalty 2846, flap count 3
*Apr 1 08:39:02.687: EvD: unsuppress item left in reuse timer array with penalty 2846
*Apr 1 08:39:02.687: BGP(0): charge penalty for 10.0.1.48/28 path 100 with half-life-
time 15 reuse/suppress 750/2000
*Apr 1 08:39:02.687: BGP(0): flapped 3 times since 00:02:33. New penalty is 2846
R2#
*Apr 1 08:39:12.402: EvD: accum. penalty decayed to 2835 after 9 second(s)
R2#
*Apr 1 08:39:32.484: BGP(0): suppress 10.0.1.48/28 path 100 for 00:28:30 (penalty 2791)
*Apr 1 08:39:32.484: half-life-time 15, reuse/suppress 750/2000
*Apr 1 08:39:32.484: EvD: accum. penalty 2791, now suppressed with a reuse intervals of 171

```

## Monitoring Route Dampening

```
router# show ip bgp dampened-paths
```

- Displays the dampened routes

```
router# show ip bgp flap-statistics [{regex regexp} | {filter-
list access-list} | {ip-address mask [longer-prefix]}]
```

- Displays flap statistics for all routes with dampening history
- Can match routes against regular expressions, AS-path access-lists, a specific route, or more specific routes

```
router# debug ip bgp dampening
```

- Displays the BGP dampening events

The penalty that is placed on a network is decayed until the reuse limit is reached, upon which the route is once again advertised. Every time that a route flap occurs, the penalty is recalculated. In the router, the penalty is encoded as the time that it takes for the penalty to decay below the reuse limit. At half of the reuse limit, the dampening information for the route to the network is removed.

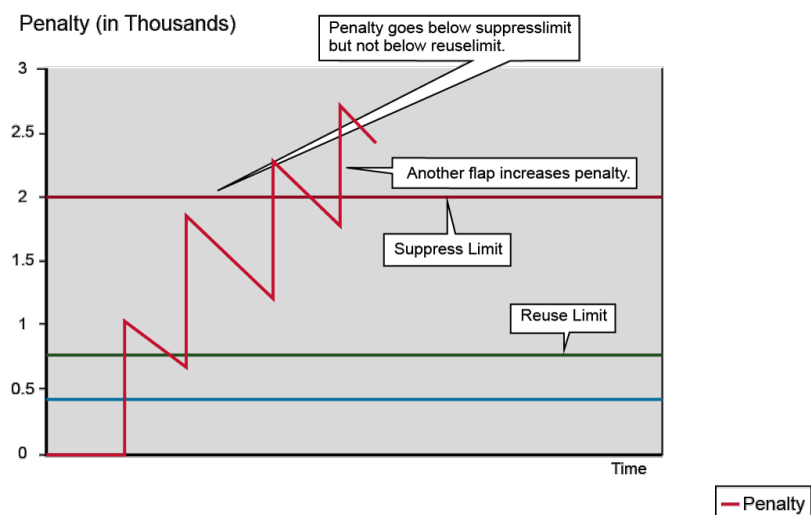
Use the **show ip bgp dampened-paths** command to list all routes that are currently suppressed because of dampening.

Use the **show ip bgp flap-statistics** command to list all routes that have a penalty that is still above the time-to-forget limit. You can also use this command in combination with regular expressions and filter-lists.

The **show ip bgp flap-statistics prefix** command displays detailed dampening information about a specific network.

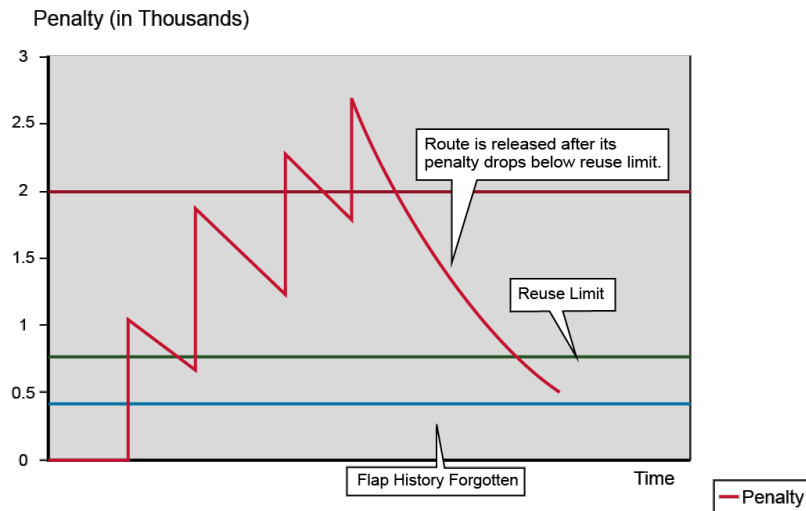
The **show ip bgp flap-statistics prefix mask longer-prefix** command displays dampening information about a specific network and its subnets.

Use the **debug ip bgp dampening** command to display BGP dampening events as they occur in real time.





The penalty of the route is decreased following an exponential curve. After a while, the penalty drops below the suppress limit, but the route is not yet released. The route is released only after the penalty drops further below the reuse limit. In the example here, the route flaps again, further increasing the penalty.



In the example here, the route has stabilized, and the penalty gradually drops below the reuse limit. At that point, the BGP router releases the route, and it can now be selected as the best BGP path. If the router selects the released route as the best BGP path, it is also propagated to BGP neighbors and used for data forwarding.

When the penalty that is associated with a route drops below half of the reuse limit, the router clears the penalty and the flap history that are associated with the route.

### Step 6

On the R2 router, verify the 10.0.1.0/48 entry in the BGP table.

```
R2# show ip bgp 10.0.1.48/28
BGP routing table entry for 10.0.1.48/28, version 47
Paths: (1 available, no best path)
 Not advertised to any peer
 Refresh Epoch 1
 100, (suppressed due to dampening)
 172.16.12.11 from 172.16.12.11 (10.0.40.1)
 Origin IGP, metric 0, localpref 100, valid, external
 Dampinfo: penalty 2674, flapped 3 times in 00:04:00, reuse in 00:05:39
```

You should see that network 10.0.1.0/48 received via 172.16.12.11 neighbor gets suppressed due to dampening.

### Step 7

On the R2 router, verify the content of the whole BGP table.

```
R2# show ip bgp
BGP table version is 47, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/24 0.0.0.0 0 32768 i
*> 10.0.0.16/28 0.0.0.0 0 32768 i
*> 10.0.0.32/28 0.0.0.0 0 32768 i
*> 10.0.0.48/28 0.0.0.0 0 32768 i
*> 10.0.0.64/28 0.0.0.0 0 32768 i
*> 10.0.1.0/28 172.16.12.11 0 0 100 i
*> 10.0.1.16/28 172.16.12.11 0 0 100 i
*> 10.0.1.32/28 172.16.12.11 0 0 100 i
*d 10.0.1.48/28 172.16.12.11 0 0 100 i
*> 10.0.1.64/28 172.16.12.11 0 0 100 i
```

```
*> 10.0.1.80/28 172.16.12.11 0 0 100 i
<... output omitted...>
```

You should see network 10.0.1.0/48 marked as dampened.

### Step 8

On the R2 router, display all dampened paths.

```
R2# show ip bgp dampening dampened-paths
BGP table version is 47, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network From Reuse Path
*d 10.0.1.48/28 172.16.12.11 00:04:49 100 i
```

You should see that network 10.0.1.0/48 marked as dampened for a path via 172.16.12.11 neighbor in autonomous system 100.

### Step 9

On the R2 router, display flapping statistics for BGP route dampening.

```
R2# show ip bgp dampening flap-statistics
BGP table version is 47, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network From Flaps Duration Reuse Path
*d 10.0.1.48/28 172.16.12.11 3 00:05:20 00:04:19 100
```

You should see that network 10.0.1.0/48 flapped three times.

### Step 10

On the R2 router, display BGP dampening parameters.

```
R2# show ip bgp dampening parameters
dampening 15 750 2000 60 (DEFAULT)
Half-life time : 15 mins Decay Time : 2320 secs
Max suppress penalty: 12000 Max suppress time: 60 mins
Suppress penalty : 2000 Reuse penalty : 750
```

You should see the default BGP dampening parameters, for instance for suppress penalty. The default value is 2000.

## Releasing Dampened Routes

### Monitoring Route Reflectors

```
router# clear ip bgp ip-address flap-
statistics [{regex regexp} | {filter-list list-name} | {ip-
address network-mask}]
```

- Clears the flap statistics but does not release dampened routes.

```
router# clear ip bgp dampening [ip-address network-mask]
```

- Releases all the dampened routes or just the specified network.
- Flap statistics or dampened routes are also cleared when the BGP session with the neighbor is lost.

There are two timers that are calculated for every route when it flaps:

- **Time to forget:** The time that it takes before all flap history is deleted. Using the **clear ip bgp flap-statistics** command deletes all penalty information, but it does not release the dampened paths.
- **Time to reuse:** The time that it takes before a route can be considered for best-path processing. Using the **clear ip bgp dampening** command resets this timer for all networks or for specified networks so that they are no longer suppressed. The flap statistics, however, are still kept, and the next flap will cause the previously dampened paths to be suppressed again.

## clear ip bgp flap-statistics

To clear BGP flap statistics, use the **clear ip bgp flap-statistics** privileged EXEC command.

- **clear ip bgp** *ip-address* **flap-statistics** [{**regexp** *regexp*} | {**filter-list** *list-name*} | {*ip-address* *network-mask*}]

### Syntax Description

| Parameter                           | Description                                                                                                                                                                                                            |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>                   | (Optional) Clears flap statistics for a single entry at this IP address. If this argument is placed before <b>flap-statistics</b> , the router clears flap statistics for all paths from the neighbor at this address. |
| <b>regexp</b> <i>regexp</i>         | (Optional) Clears flap statistics for all the paths that match the regular expression.                                                                                                                                 |
| <b>filter-list</b> <i>list-name</i> | (Optional) Clears flap statistics for all the paths that pass the access-list.                                                                                                                                         |
| <i>network-mask</i>                 | (Optional) Network mask that is applied to the <i>ip-address</i> argument.                                                                                                                                             |

## clear ip bgp dampening

To clear BGP route-dampening information and unsuppress the suppressed routes, use the **clear ip bgp dampening** privileged EXEC command.

- **clear ip bgp dampening** [*ip-address* *network-mask*]

### Syntax Description

| Parameter           | Description                                                                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>ip-address</i>   | (Optional) Clears flap statistics for a single entry at this IP address. If this argument is placed before <b>flap-statistics</b> , the router clears flap statistics for all paths from the neighbor at this address. |
| <i>network-mask</i> | (Optional) Network mask that is applied to the <i>ip-address</i> argument.                                                                                                                                             |

### Step 11

Clear the BGP dampening information on the R2 router.

```
R2# clear ip bgp dampening
```

### Step 12

On the R2 router, verify the 10.0.1.0/48 entry in the BGP table.

```
R2# show ip bgp 10.0.1.48/28
BGP routing table entry for 10.0.1.48/28, version 48
Paths: (1 available, best #1, table default)
 Advertised to update-groups:
 2
 Refresh Epoch 1
 100
 172.16.12.11 from 172.16.12.11 (10.0.40.1)
 Origin IGP, metric 0, localpref 100, valid, external, best
```

You should see that network 10.0.1.0/48 received via 172.16.12.11 neighbor is not suppressed anymore.

### Step 13

On the R2 router, verify the content of the whole BGP table.

```
R2# show ip bgp
BGP table version is 48, local router ID is 10.0.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
 r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
 x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found

 Network Next Hop Metric LocPrf Weight Path
*> 10.0.0.0/24 0.0.0.0 0 32768 i
*> 10.0.0.16/28 0.0.0.0 0 32768 i
*> 10.0.0.32/28 0.0.0.0 0 32768 i
*> 10.0.0.48/28 0.0.0.0 0 32768 i
*> 10.0.0.64/28 0.0.0.0 0 32768 i
*> 10.0.1.0/28 172.16.12.11 0 0 100 i
*> 10.0.1.16/28 172.16.12.11 0 0 100 i
*> 10.0.1.32/28 172.16.12.11 0 0 100 i
*> 10.0.1.48/28 172.16.12.11 0 0 100 i
*> 10.0.1.64/28 172.16.12.11 0 0 100 i
*> 10.0.1.80/28 172.16.12.11 0 0 100 i
<...output omitted...>
```

You should see that network 10.0.1.0/48 is not marked as dampened anymore.

## Summary

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

- Route dampening is a BGP feature that is designed to reduce BGP processing requirements by minimizing the propagation of unstable routes to BGP peers.
- A router with route dampening enabled keeps track of all routes and the penalties that are assigned to them.
- Each time a flap occurs, the flapping route receives 1000 penalty points.
- If the route penalties reach the suppress limit, the route is no longer forwarded to other neighbors until the assigned penalty has decayed below the reuse limit.
- You can implement route dampening with the **bgp dampening** command either globally in the BGP process or selectively using a route-map.
- Use the **clear ip bgp flap-statistics** command to delete all penalty information without releasing the dampened paths.
- The **clear ip bgp dampening** command clears dampening information and releases suppressed routes.
- The **show ip bgp dampened-paths** command lists all routes that are currently suppressed because of dampening, the

# Module Summary

## Overview

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

- Increased BGP convergence time and high CPU utilization caused by the BGP scanner and BGP router processes can be reduced with Cisco IOS tools such as PMTU discovery, BGP PIC, BFD, as well as by adjusting BGP timers and intervals..
- The maximum-prefix function is a scalable solution that limits the number of routes that a BGP router can receive from a specific neighbor.
- Peer groups are a fundamental BGP scalability tool and should be used in all environments where a router has a large number of BGP neighbors.
- Route dampening is a BGP feature that is designed to reduce BGP processing requirements by minimizing the propagation of unstable routes to BGP peers.

## References

For additional information, refer to these resources:

- Cisco Systems, Inc. *BGP Restart Neighbor Session After Maximum-Prefix Limit Reached*.  
[http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute\\_bgp/configuration/15-mt/irg-15-mt-book/BGP-Sub-Codes-for-BGP-Cease-Notification.pdf](http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/15-mt/irg-15-mt-book/BGP-Sub-Codes-for-BGP-Cease-Notification.pdf)
- Cisco Systems, Inc. *BGP Peer Groups*.  
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13755-29.html>
- Cisco Systems, Inc. *BGP Case Studies* "BGP Case Studies 4."  
<http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/26634-bgp-toc.html#case4>

## 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. What are three characteristics of a converged BGP network?  
(Choose three.)

- ☐ The input queue and output queue for all peers is 0.
- ☐ All routes in the BGP table have been installed in the routing table.
- ☐ The table version for all peers equals the table version of the BGP table.
- ☐ All routes have been accepted.

1. Which two of the following modifications result in improved BGP convergence? (Choose two.) (Source: "Improving BGP Convergence")

- ☐ Increasing the default value of BGP hold time.
- ☐ Lowering the default value of BGP scan time.
- ☐ Increasing the default value of the neighbor advertisement intervals.
- ☐ Lowering the default value of the neighbor advertisement intervals.

1. What is the main task of the BGP scanner process? (Source: "Improving BGP Convergence")

- Sends routing updates to BGP neighbors.
- Walks the BGP table for routes to enter into the IP routing table.
- Confirms the reachability of BGP next hops.
- Scans the router configuration to establish and maintain BGP neighbors.

1. One of your BGP core routers is experiencing periodic slow responses to ping packets that are being directed to it from the network management console. The router has just been configured to receive full Internet routes, and you suspect that the BGP router process is causing CPU utilization issues in the core router. Which two router commands should you use to confirm your suspicion? (Choose two.) (Source: "Improving BGP Convergence")

☐ **show ip route**

☐ **show ip bgp summary**

☐ **show process cpu**

☐ **show memory**

1. Look at the output of the **show interfaces Ethernet0/0** command. How has the interface been modified to improve BGP convergence? (Source: "Improving BGP Convergence")  
router# **show interfaces Ethernet0/0**  
Ethernet0/0 is up, line protocol is up  
Hardware is DEC21140, address is 0000.0c0c.1111 (bia 0002.eaa3.5a60)  
Internet address is 112.64.101.17 255.255.255.240  
MTU 1460 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 200/255  
Encapsulation ARPA, loopback not set, keepalive not set, hdx, 100BaseTX  
ARP type: ARPA, ARP Timeout 4:00:00  
Last input never, output 0:00:16, output hang 0:28:01  
Last clearing of "show interface" counters 0:20:05  
Output queue 25/40, 0 drops; input queue 50/500, 1470 drops  
5 minute input rate 21666400 bits/sec, 1855 packets/sec  
5 minute output rate 72221 bits/sec, 618 packets/sec

The output queue has been decreased to expedite packet forwarding out the Fast Ethernet interface.

The drop threshold of the input queue has been set to begin randomly discarding packets after the queue reaches 50 packets deep.

PMTU discovery has been enabled by setting the interface MSS to 1460 bytes.

PMTU discovery has been enabled by setting the interface MSS to 1460 bytes.

1. Refer to the output. Which two parameters would indicate that the BGP network has converged? (Choose two.) (Source: "Improving BGP Convergence")
- ```
router# show ip bgp summary
BGP router identifier 172.16.0.4, local AS number 1
BGP table version is 16, main routing table version 16
20 network entries and 20 paths using 2826 bytes of memory
8 BGP path attribute entries using 480 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
3 BGP community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
3 BGP filter-list cache entries using 36 bytes of memory
BGP activity 20/0 prefixes, 24/4 paths, scan interval 120 secs
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ OutQ
Up/Down State/PfxRcd
172.16.0.1    4    1    30     30     16   0   0 00:23:13    5
172.16.0.2    4    1    33     30     16   0   0 00:23:15    5
172.16.0.3    4    1    27     30     16   0   0 00:23:14    5
192.168.21.99 4   99    31     35     16   0   0 00:23:04    5
```

☐ The TblVer for all neighbors is 16.

☐ V is set to 4 for all neighbors.

☐ The InQ and OutQ for all neighbors is 0.

☐ All neighbors are in the Established state and have the same PfxRcd value.

1. Refer to the output. How frequently the BGP scanner process will run on the router? (Source: "Improving BGP Convergence")router#

show ip bgp summary

```
BGP router identifier 172.16.0.4, local AS number 1
BGP table version is 16, main routing table version 16
20 network entries and 20 paths using 2826 bytes of memory
8 BGP path attribute entries using 480 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
3 BGP community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
3 BGP filter-list cache entries using 36 bytes of memory
BGP activity 20/0 prefixes, 24/4 paths, scan interval 120 secs
Neighbor      V   AS MsgRcvd MsgSent  TblVer  InQ OutQ
Up/Down State/PfxRcd
172.16.0.1    4    1    30     30     16   0   0 00:23:13    5
172.16.0.2    4    1    33     30     16   0   0 00:23:15    5
172.16.0.3    4    1    27     30     16   0   0 00:23:14    5
192.168.21.99 4   99    31     35     16   0   0 00:23:04    5
```

By default, the process will run every 60 seconds.

The process has run 16 times and will run again when the next BGP update arrives.

The process will run on this router every 120 seconds.

It cannot be determined from this output.

1. What are two potential issues that are caused by modifying the default scan time and advertisement interval on a BGP router? (Choose two.) (Source: "Improving BGP Convergence")

- ☐ Router CPU resources can be exhausted.
- ☐ Router memory resources can be depleted.
- ☐ Routing loops are more likely.
- ☐ BGP could converge faster than the IGP and cause network black holes.

1. Which option can be used to support traffic forwarding during RP switchover? (Source: "Improving BPG Convergence")

- ☐ Nonstop Forwarding
- ☐ Cisco Express Forwarding
- ☐ Perfect Forwarding Routing

1. Lowering the keepalive timer between sent keepalive messages to a BGP neighbor will improve BGP convergence. True or false? (Source: "Improving BGP Convergence")

☒ True

☐ False

1. What are three reasons to limit the number of BGP prefixes that are received from a neighbor? (Choose three.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

- ☐ To prevent denial-of-service attacks.
- ☐ To protect against incorrect router configuration on the neighbor side.
- ☐ To prevent redundant routing information from being loaded into the BGP table.
- ☐ To avoid overloading router memory and CPU resources.

1. In which two situations would a directly connected BGP neighbor stay in the Idle state? (Choose two.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

☐ The neighbor has exceeded the maximum number of allowed prefixes.

☐ The maximum-prefix threshold has been reached.

☐ The **restart** option has not been specified with the **maximum-prefix** command.

☐ The neighbor is more than one hop away.

1. Which two of the following characteristics accurately describe the **show ip bgp neighbors** command? (Choose two.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

☐ For neighbors with the maximum-prefix function configured, displays the maximum number of prefixes and the warning threshold.

☐ For neighbors exceeding the maximum number of prefixes, displays the reason that the BGP session is idle.

☐ For neighbors with unstable routes, displays the feasible successor for those routes.

☐ For neighbors in confederations, displays the route reflector status of those neighbors.

1. What is the need for BGP peer groups? (Source: "Implementing BGP Peer Groups")

- Can be used to configure the same set of parameters for a number of BGP neighbors in a common template.

- Can be used to allow anonymous BGP neighbors.

- Allow EBGP peers to be configured with the same AS number and parameters.

- Can be used to hide the identity of BGP peers from external neighbors.

1. Which of the following statements about the benefit of BGP peer groups is accurate? (Source: "Implementing BGP Peer Groups")

☐ With BGP peer groups, all of the router CPU utilization that is imposed by BGP update generation is significantly reduced.

☐ With BGP peer groups, some of the router CPU utilization that is imposed by BGP update generation is significantly reduced.

☐ Network administrators should use peer groups to make smaller networks more productive.

☐ With BGP peer groups, neighbor relationships are automatically created.

1. What are two limitations of BGP peer groups on Cisco routers? (Choose two.) (Source: "Implementing BGP Peer Groups")

☐ EBGP and IBGP neighbors cannot be members of the same peer group.

☐ All routers in the peer group must belong to the same AS.

☐ Peer group members cannot contain different outbound filtering mechanisms.

☐ Peer group members must have the same inbound filtering mechanisms.

1. Which two of the following characteristics accurately describe BGP peer groups? (Choose two.) (Source: "Implementing BGP Peer Groups")

- ☐ A BGP peer group creates a neighbor parameter template.
- ☐ When actual neighboring routers are assigned to the peer group on a router, all of the attributes that are configured for the peer group are applied to selected peer group members.
- ☐ One of the configurable parameters includes community propagation.
- ☐ Individual parameters specified in a peer group cannot be overridden on a neighbor-by-neighbor basis.

1. Which two of the following characteristics accurately describe the function of the BGP Dynamic Update Peer-Groups feature? (Choose two.) (Source: "Implementing BGP Peer Groups")

☐ Does not provide the operator with time to change the configuration if a mistake is made.

☐ Separates BGP update generation from peer group configuration.

☐ Does not require any configuration by the network operator.

☐ Requires minimal configuration by the network operator.

1. Which two of the following statements accurately describe the function of the BGP peer templates feature? (Choose two.) (Source: "Implementing BGP Peer Groups")

- ☐ Network operators must still configure some peer groups in BGP, even if using the BGP Configuration Using Peer Templates feature.
- ☐ Peer templates overcome all limitations of peer groups.
- ☐ Peer templates improve the flexibility and enhance the capability of neighbor configuration.
- ☐ You can chain together peer template configurations to create simple or complex configurations.

1. What are three steps that are required to properly configure BGP peer groups on Cisco routers? (Choose three.) (Source: "Implementing BGP Peer Groups")

- ☐ Specify parameters for the BGP peer group
- ☐ Create a BGP peer group
- ☐ Enable the peer group by clearing the BGP session
- ☐ Assign a neighbor into the peer group

1. Which command do you use to display the summary status of all neighbors in a peer group? (Source: "Implementing BGP Peer Groups")

☐ **show ip bgp**

☐ **show peer-group summary**

☐ **show ip bgp neighbor**

☐ **show ip bgp peer-group summary**

1. Which two descriptions of the purpose of BGP route dampening are accurate? (Choose two.) (Source: "Using BGP Route Dampening")

- ☐ A tool designed to help minimize the number of BGP updates
- ☐ Suppresses routes that occasionally flap
- ☐ Designed to reduce router processing load caused by unstable routes
- ☐ Prevents sustained routing oscillation, with some effect on other well-behaved routes

1. Which two mechanisms are built into BGP to make it more scalable by reducing the route-processing requirements of BGP routers? (Choose two.) (Source: "Using BGP Route Dampening")

- ☐ Split horizon
- ☐ Route dampening
- ☐ Synchronization
- ☐ Per-neighbor update timers

1. What are two things that happen to an EBGp route that has become unreachable when BGP route dampening is used? (Choose two.)
(Source: "Using BGP Route Dampening")

- ☐ It is removed from the IP routing table.
- ☐ It is removed from the BGP table.
- ☐ It remains in the IP routing table as long as its penalty remains greater than 50 percent of the reuse limit.
- ☐ It is kept in the BGP table and marked as a history entry.

1. What are two methods of enabling route dampening on a Cisco router? (Choose two.) (Source: "Using BGP Route Dampening", Topic: "Configuring BGP Route Dampening")

☐ Globally, by enabling route dampening in global router configuration mode

☐ Globally, by enabling route dampening under the BGP routing process

☐ On specific routes by enabling route dampening on a specific interface

☐ By using a route-map in the BGP process to apply route dampening to specific routes

1. Which two things could happen to a BGP route that is penalized above the reuse limit but has an assigned penalty that is under the suppress limit? (Choose two.) (Source: "Using BGP Route Dampening")

☐ The route is suppressed from BGP updates if it is reachable.

☐ The route is marked as a history entry in the BGP table.

☐ The route is withdrawn from the IP routing table.

☐ The route continues to be advertised.

Answer Key

1. What are three characteristics of a converged BGP network?
(Choose three.)

- ☒ The input queue and output queue for all peers is 0.
- ☐ All routes in the BGP table have been installed in the routing table.
- ☒ The table version for all peers equals the table version of the BGP table.
- ☒ All routes have been accepted.

1. Which two of the following modifications result in improved BGP convergence? (Choose two.) (Source: "Improving BGP Convergence")

- ☐ Increasing the default value of BGP hold time.
- ☒ Lowering the default value of BGP scan time.
- ☐ Increasing the default value of the neighbor advertisement intervals.
- ☒ Lowering the default value of the neighbor advertisement intervals.

1. What is the main task of the BGP scanner process? (Source: "Improving BGP Convergence")

- Sends routing updates to BGP neighbors.

- Walks the BGP table for routes to enter into the IP routing table.

- Confirms the reachability of BGP next hops.

- Scans the router configuration to establish and maintain BGP neighbors.

1. One of your BGP core routers is experiencing periodic slow responses to ping packets that are being directed to it from the network management console. The router has just been configured to receive full Internet routes, and you suspect that the BGP router process is causing CPU utilization issues in the core router. Which two router commands should you use to confirm your suspicion? (Choose two.) (Source: "Improving BGP Convergence")

☐ **show ip route**

☒ **show ip bgp summary**

☒ **show process cpu**

☐ **show memory**

1. Look at the output of the **show interfaces Ethernet0/0** command. How has the interface been modified to improve BGP convergence? (Source: "Improving BGP Convergence")
router# **show interfaces Ethernet0/0**
Ethernet0/0 is up, line protocol is up
Hardware is DEC21140, address is 0000.0c0c.1111 (bia 0002.eaa3.5a60)
Internet address is 112.64.101.17 255.255.255.240
MTU 1460 bytes, BW 100000 Kbit, DLY 100 usec, rely 255/255, load 200/255
Encapsulation ARPA, loopback not set, keepalive not set, hdx, 100BaseTX
ARP type: ARPA, ARP Timeout 4:00:00
Last input never, output 0:00:16, output hang 0:28:01
Last clearing of "show interface" counters 0:20:05
Output queue 25/40, 0 drops; input queue 50/500, 1470 drops
5 minute input rate 21666400 bits/sec, 1855 packets/sec
5 minute output rate 72221 bits/sec, 618 packets/sec

The output queue has been decreased to expedite packet forwarding out the Fast Ethernet interface.

The drop threshold of the input queue has been set to begin randomly discarding packets after the queue reaches 50 packets deep.

PMTU discovery has been enabled by setting the interface MSS to 1460 bytes.

PMTU discovery has been enabled by setting the interface MSS to 1460 bytes.

1. Refer to the output. Which two parameters would indicate that the BGP network has converged? (Choose two.) (Source: "Improving BGP Convergence")
- ```
router# show ip bgp summary
BGP router identifier 172.16.0.4, local AS number 1
BGP table version is 16, main routing table version 16
20 network entries and 20 paths using 2826 bytes of memory
8 BGP path attribute entries using 480 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
3 BGP community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
3 BGP filter-list cache entries using 36 bytes of memory
BGP activity 20/0 prefixes, 24/4 paths, scan interval 120 secs
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ
Up/Down State/PfxRcd
172.16.0.1 4 1 30 30 16 0 0 00:23:13 5
172.16.0.2 4 1 33 30 16 0 0 00:23:15 5
172.16.0.3 4 1 27 30 16 0 0 00:23:14 5
192.168.21.99 4 99 31 35 16 0 0 00:23:04 5
```

- ☒ The TblVer for all neighbors is 16.
- ☐ V is set to 4 for all neighbors.
- ☒ The InQ and OutQ for all neighbors is 0.
- ☐ All neighbors are in the Established state and have the same PfxRcd value.



1. Refer to the output. How frequently the BGP scanner process will run on the router? (Source: "Improving BGP Convergence")router#

**show ip bgp summary**

```
BGP router identifier 172.16.0.4, local AS number 1
BGP table version is 16, main routing table version 16
20 network entries and 20 paths using 2826 bytes of memory
8 BGP path attribute entries using 480 bytes of memory
7 BGP AS-PATH entries using 168 bytes of memory
3 BGP community entries using 72 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
3 BGP filter-list cache entries using 36 bytes of memory
BGP activity 20/0 prefixes, 24/4 paths, scan interval 120 secs
Neighbor V AS MsgRcvd MsgSent TblVer InQ OutQ
Up/Down State/PfxRcd
172.16.0.1 4 1 30 30 16 0 0 00:23:13 5
172.16.0.2 4 1 33 30 16 0 0 00:23:15 5
172.16.0.3 4 1 27 30 16 0 0 00:23:14 5
192.168.21.99 4 99 31 35 16 0 0 00:23:04 5
```

By default, the process will run every 60 seconds.

The process has run 16 times and will run again when the next BGP update arrives.

The process will run on this router every 120 seconds.

It cannot be determined from this output.

1. What are two potential issues that are caused by modifying the default scan time and advertisement interval on a BGP router? (Choose two.) (Source: "Improving BGP Convergence")

- ☒ Router CPU resources can be exhausted.
- ☒ Router memory resources can be depleted.
- ☐ Routing loops are more likely.
- ☐ BGP could converge faster than the IGP and cause network black holes.

1. Which option can be used to support traffic forwarding during RP switchover? (Source: "Improving BPG Convergence")

|                                                  |
|--------------------------------------------------|
| <input type="radio"/> Nonstop Forwarding         |
| <input type="radio"/> Cisco Express Forwarding   |
| <input type="radio"/> Perfect Forwarding Routing |

1. Lowering the keepalive timer between sent keepalive messages to a BGP neighbor will improve BGP convergence. True or false? (Source: "Improving BGP Convergence")

☒ True

☐ False

1. What are three reasons to limit the number of BGP prefixes that are received from a neighbor? (Choose three.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

- ☒ To prevent denial-of-service attacks.
- ☒ To protect against incorrect router configuration on the neighbor side.
- ☐ To prevent redundant routing information from being loaded into the BGP table.
- ☒ To avoid overloading router memory and CPU resources.

1. In which two situations would a directly connected BGP neighbor stay in the Idle state? (Choose two.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

☒ The neighbor has exceeded the maximum number of allowed prefixes.

☐ The maximum-prefix threshold has been reached.

☒ The **restart** option has not been specified with the **maximum-prefix** command.

☐ The neighbor is more than one hop away.

1. Which two of the following characteristics accurately describe the **show ip bgp neighbors** command? (Choose two.) (Source: "Limiting the Number of Prefixes Received from a BGP Neighbor")

☒ For neighbors with the maximum-prefix function configured, displays the maximum number of prefixes and the warning threshold.

☒ For neighbors exceeding the maximum number of prefixes, displays the reason that the BGP session is idle.

☐ For neighbors with unstable routes, displays the feasible successor for those routes.

☐ For neighbors in confederations, displays the route reflector status of those neighbors.

1. What is the need for BGP peer groups? (Source: "Implementing BGP Peer Groups")

- Can be used to configure the same set of parameters for a number of BGP neighbors in a common template.

- Can be used to allow anonymous BGP neighbors.

- Allow EBGP peers to be configured with the same AS number and parameters.

- Can be used to hide the identity of BGP peers from external neighbors.



1. Which of the following statements about the benefit of BGP peer groups is accurate? (Source: "Implementing BGP Peer Groups")

☐ With BGP peer groups, all of the router CPU utilization that is imposed by BGP update generation is significantly reduced.

☐ With BGP peer groups, some of the router CPU utilization that is imposed by BGP update generation is significantly reduced.

☐ Network administrators should use peer groups to make smaller networks more productive.

☐ With BGP peer groups, neighbor relationships are automatically created.

1. What are two limitations of BGP peer groups on Cisco routers? (Choose two.) (Source: "Implementing BGP Peer Groups")

☒ EBGP and IBGP neighbors cannot be members of the same peer group.

☐ All routers in the peer group must belong to the same AS.

☒ Peer group members cannot contain different outbound filtering mechanisms.

☐ Peer group members must have the same inbound filtering mechanisms.

1. Which two of the following characteristics accurately describe BGP peer groups? (Choose two.) (Source: "Implementing BGP Peer Groups")

☒ A BGP peer group creates a neighbor parameter template.

☐ When actual neighboring routers are assigned to the peer group on a router, all of the attributes that are configured for the peer group are applied to selected peer group members.

☒ One of the configurable parameters includes community propagation.

☐ Individual parameters specified in a peer group cannot be overridden on a neighbor-by-neighbor basis.

1. Which two of the following characteristics accurately describe the function of the BGP Dynamic Update Peer-Groups feature? (Choose two.) (Source: "Implementing BGP Peer Groups")

☐ Does not provide the operator with time to change the configuration if a mistake is made.

☒ Separates BGP update generation from peer group configuration.

☒ Does not require any configuration by the network operator.

☐ Requires minimal configuration by the network operator.

1. Which two of the following statements accurately describe the function of the BGP peer templates feature? (Choose two.) (Source: "Implementing BGP Peer Groups")

☐ Network operators must still configure some peer groups in BGP, even if using the BGP Configuration Using Peer Templates feature.

☐ Peer templates overcome all limitations of peer groups.

☒ Peer templates improve the flexibility and enhance the capability of neighbor configuration.

☒ You can chain together peer template configurations to create simple or complex configurations.

1. What are three steps that are required to properly configure BGP peer groups on Cisco routers? (Choose three.) (Source: "Implementing BGP Peer Groups")

- ☒ Specify parameters for the BGP peer group
- ☒ Create a BGP peer group
- ☐ Enable the peer group by clearing the BGP session
- ☒ Assign a neighbor into the peer group

1. Which command do you use to display the summary status of all neighbors in a peer group? (Source: "Implementing BGP Peer Groups")

☐ **show ip bgp**

☐ **show peer-group summary**

☐ **show ip bgp neighbor**

☐ **show ip bgp peer-group summary**

1. Which two descriptions of the purpose of BGP route dampening are accurate? (Choose two.) (Source: "Using BGP Route Dampening")



A tool designed to help minimize the number of BGP updates



Suppresses routes that occasionally flap



Designed to reduce router processing load caused by unstable routes



Prevents sustained routing oscillation, with some effect on other well-behaved routes



1. Which two mechanisms are built into BGP to make it more scalable by reducing the route-processing requirements of BGP routers? (Choose two.) (Source: "Using BGP Route Dampening")

- ☐ Split horizon
- ☒ Route dampening
- ☐ Synchronization
- ☒ Per-neighbor update timers

1. What are two things that happen to an EBGp route that has become unreachable when BGP route dampening is used? (Choose two.)  
(Source: "Using BGP Route Dampening")

- ☒ It is removed from the IP routing table.
- ☐ It is removed from the BGP table.
- ☐ It remains in the IP routing table as long as its penalty remains greater than 50 percent of the reuse limit.
- ☒ It is kept in the BGP table and marked as a history entry.

1. What are two methods of enabling route dampening on a Cisco router? (Choose two.) (Source: "Using BGP Route Dampening", Topic: "Configuring BGP Route Dampening")

☐ Globally, by enabling route dampening in global router configuration mode

☒ Globally, by enabling route dampening under the BGP routing process

☐ On specific routes by enabling route dampening on a specific interface

☒ By using a route-map in the BGP process to apply route dampening to specific routes

1. Which two things could happen to a BGP route that is penalized above the reuse limit but has an assigned penalty that is under the suppress limit? (Choose two.) (Source: "Using BGP Route Dampening")

- ☐ The route is suppressed from BGP updates if it is reachable.
- ☒ The route is marked as a history entry in the BGP table.
- ☒ The route is withdrawn from the IP routing table.
- ☐ The route continues to be advertised.

## Glossary

### ACK

acknowledgment. Notification sent from one network device to another to acknowledge that some event occurred (for example, the receipt of a message).

### AD

administrative distance. Rating of the trustworthiness of a routing information source. Administrative distance often is expressed as a numerical value between 0 and 255. The higher the value, the lower the trustworthiness rating.

### AFI

Address Family Identifier.

### ARIN

American Registry for Internet Numbers. A nonprofit organization that administers and registers IP numbers for the geographical areas that are currently managed by Network Solutions (InterNIC). Those areas include, but are not limited to, North America, South America, South Africa, and the Caribbean.

### ARP

Address Resolution Protocol. Internet protocol that is used to map an IP address to a MAC address. Defined in RFC 826.

### 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.

### ATM

Asynchronous Transfer Mode. The international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media, such as E3, SONET, and T3.

### BFD

Bidirectional Forwarding Detection. A detection protocol that is designed to provide fast forwarding-path failure detection times for media types, encapsulations, topologies, and routing protocols.

### BGP

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

### BGP4

Border Gateway Protocol version 4.

### CE

customer edge. Identifies the network devices, connected to a provider network, that are under the administrative control of the customer.

### CLI

command-line interface. An interface that allows the user to interact with the operating system by entering commands and optional arguments. The UNIX operating system and DOS provide CLIs.

### CPU

central processing unit. The hardware within a computer system or smartphone that carries out the instructions of a computer program by performing the basic arithmetical, logical, and input-output operations of the system.

### DF

Do not Fragment

### DMVPN

Dynamic Multipoint VPN.

### DMZ

demilitarized zone.

### DNS

Domain Name System. System used on the Internet for translating names of network nodes into addresses.

### DoS

denial of service. An intentional or unintentional attack on a device that makes the resource unavailable to perform its normal function.

### DPT

Dynamic Packet Transport

### DSL

digital subscriber line. Public network technology that delivers high bandwidth over conventional copper wiring at limited distances. There are four types of DSL: ADSL, HDSL, SDSL, and VDSL. All are provisioned via modem pairs, with one modem located at a central office and the other at the customer site. Because most DSL technologies do not use the whole bandwidth of the twisted pair, there is room remaining for a voice channel.

### EBGP

Exterior Border Gateway Protocol

## EGP

Exterior Gateway Protocol. It's the Internet protocol for exchanging routing information between autonomous systems. Documented in RFC 904. This is not to be confused with the general term *exterior gateway protocol*. EGP is an obsolete protocol that was replaced by BGP.

## EIGRP

Enhanced Interior Gateway Routing Protocol. It's the advanced version of IGRP developed by Cisco. It provides superior convergence properties and operating efficiency, and it combines the advantages of link-state protocols with those of distance vector protocols.

## Ethernet

Baseband LAN specification invented by Xerox Corporation and developed jointly by Xerox, Intel, and Digital Equipment Corporation. Ethernet networks use CSMA/CD and run over a variety of cable types at 10 Mbps. Ethernet is similar to the IEEE 802.3 series of standards. It is the most commonly used LAN technology because its protocol is easy to understand, implement, manage, and maintain. It allows low-cost network implementations, provides extensive topological flexibility for network installation, and guarantees successful interconnection and operation of standards-compliant products, regardless of manufacturer.

## Fast Ethernet

Any of a number of 100-Mbps Ethernet specifications. Fast Ethernet offers a speed increase 10 times that of the 10BaseT Ethernet specification while preserving such qualities as frame format, MAC mechanisms, and MTU. Such similarities allow the use of existing 10BaseT applications and network management tools on Fast Ethernet networks. Based on an extension to the IEEE 802.3 specification.

## FDDI

Fiber Distributed Data Interface. LAN standard, defined by ANSI X3T9.5, specifying a 100-Mbps token-passing network using fiber-optic cable, with transmission distances of up to 2 km. FDDI uses a dual-ring architecture to provide redundancy.

## FTTx

Fiber From The "X", where X stands for different type of deployment (Building, Home, Premises, etc.)

## FIB

forwarding information base.

## Frame Relay

Industry-standard, packet-switched data link layer protocol that handles multiple virtual circuits between connected devices.

## Gigabit Ethernet

Standard for a high-speed Ethernet, approved by the IEEE (Institute of Electrical and Electronics Engineers) 802.3z standards committee in 1996.

## IANA

Internet Assigned Numbers Authority. Organization operated under the auspices of the ISOC as a part of the IAB. IANA delegates authority for IP address-space allocation and domain-name assignment to the InterNIC and other organizations. IANA also maintains a database of assigned protocol identifiers that is used in the TCP/IP stack, including autonomous system numbers.

## IBGP

Internal Border Gateway Protocol.

## ICMP

Internet Control Message Protocol. Network layer Internet protocol that reports errors and provides other information that is relevant to IP packet processing. Documented in RFC 792.

## IGP

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

## IOL

Special build of Cisco IOS Software for Linux, created specially for virtualized environments.

## 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

A 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.

## LAN

local-area network. A high-speed, low-error data network covering a relatively small geographic area (up to a few thousand meters). LANs connect workstations, peripherals, terminals, and other devices in a single building or other geographically limited area. LAN standards specify cabling and signaling at the physical and data link layers of the OSI model. Ethernet, FDDI, and Token Ring are widely used LAN technologies.

## LSA

link-state advertisement. A broadcast packet used by link-state protocols that contains information about neighbors and path costs. LSAs are used by the receiving routers to maintain their routing tables. Sometimes called an LSP.

## LSP

label-switched path.

## MAC

Media Access Control. The lower of the two sublayers of the data link layer that is defined by the IEEE. The MAC sublayer handles access to shared media, such as whether token passing or contention will be used.

## MAC address

a standardized data link layer address that is required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. A MAC address is 6 bytes long and is controlled by the IEEE. It is also known as a hardware address, MAC layer address, and physical address.

## MD5

Message Digest 5. A one-way hashing algorithm that produces a 128-bit hash. Both MD5 and Secure Hash Algorithm (SHA) are variations on MD4 and are designed to strengthen the security of the MD4 hashing algorithm. Cisco uses hashes for authentication within the IPsec framework. MD5 is also used for message authentication in SNMP v2. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

## MED

multi-exit discriminator.

## MPLS

Multiprotocol Label Switching. a switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on pre-established IP routing information.

## MSS

maximum segment size. A TCP parameter that specifies the maximum amount of data that a device can receive in a single TCP segment.

## MTU

maximum transmission unit. The maximum packet size, in bytes, that a particular interface can handle.

## NAT

Network Address Translation. A mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating these addresses into globally routable address space. Also known as Network Address Translator.

## NHT

Next hop tracking

## NLRI

Network Layer Reachability Information.

## NSF

Nonstop Forwarding.

## ORF

outbound route filter.

## 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.

## PA

provider-assigned.

## PA address

provider-assigned address.

## PE

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

## PI

provider-independent.

## PI address

provider-independent address.

## **PIC**

Prefix-Independent Convergence.

## **PMTU**

path maximum transmission unit.

## **P-network**

provider network.

## **POI**

point of insertion.

## **POP**

Post Office Protocol. Protocol that client email applications use to retrieve mail from a mail server.

## **P router**

provider router.

## **PVC**

permanent virtual circuit (or connection). Virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and tear down in situations where certain virtual circuits must exist all the time. In ATM terminology, called a permanent virtual connection.

## **QoS**

quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

## **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.

## **RIB**

routing information base.

## **RIP**

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

## **RIPE**

Reseaux IP Europeens. Group formed to coordinate and promote TCP/IP-based networks in Europe.

## **RP**

route processor.

## **RST**

reset. A type of message.

## **RT**

route target.

## **SAFI**

Subsequent Address Family Identifier.

## **SOO**

Site of Origin.

## **SPD**

Selective Packet Discard.

## **SSO**

Stateful Switchover.

## **SYN**

synchronization.

## **SYN-ACK**

synchronization-acknowledgment.

## **TCP**

Transmission Control Protocol. Connection-oriented transport layer protocol that provides reliable full-duplex data transmission. TCP is part of the TCP/IP protocol stack.

## **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.

## **TLV**

type, length, value.

## **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.

## VPN

virtual private network. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses tunneling to encrypt all information at the IP level.

## VPNv4

virtual private network version 4. Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another. A VPN uses tunneling to encrypt all information at the IP level.

## VRF

a VPN routing/forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE router.

## 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.