BGP

Configuring BGP on Cisco Routers

Version 3.1

Lab Guide

Text Part Number: 97-1891-01

BGP

Lab Guide

Overview

Use the exercises here to complete the lab activities for this course. The solutions information is found in the Lab Exercise Answer Key.

Outline

This Lab Guide includes these exercises:

- Lab Exercise 1-1: Initial Lab Setup

- Lab Exercise 1-2: Configuring Basic BGP

- Lab Exercise 2-1: Configuring a Transit AS

- Lab Exercise 3-1: Using Multihomed BGP Networks

- Lab Exercise 3-2: Employing AS-Path Filters

- Lab Exercise 3-3: Filtering with Prefix-Lists

- Lab Exercise 3-4: Implementing Changes in BGP Policy

- Lab Exercise 4-1: Influencing BGP Route Selection with Weights

- Lab Exercise 4-2: Setting BGP Local Preference

- Lab Exercise 4-3: Understanding BGP Multi-Exit Discriminators

- Lab Exercise 4-4: Addressing BGP Communities

- Lab Exercise 6-1: Introducing Route Reflectors

- Lab Exercise 6-2: Configuring and Monitoring Confederations

- Lab Exercise 7-1: Limiting the Number of Prefixes Received from a BGP Neighbor

- Lab Exercise 7-2: Implementing BGP Peer Groups

- Lab Exercise 7-3: Using BGP Route Dampening

Lab Exercise 1-1: Initial Lab Setup

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will perform initial router configuration, configure the interfaces on your routers, and establish IGP connectivity across your core backbone. After completing this exercise, you will be able to meet these objectives:

■ Prepare your BGP student workgroup to complete the BGP lab exercises in the *Configuring BGP on Cisco Routers* (BGP) v3.1 course

Required Resources

These are the resources and equipment required to complete this exercise:

■ Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

■ Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

■ Two Ethernet 10BASE-T patch cables.

■ IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

■ One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

■ Two Cisco 2610 routers with BGP-capable operating system software installed

■ One Cisco 3640 router with an installed NM-8A/S

■ Two Catalyst 2924M-XL Ethernet switches

■ Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
interface type *number*	Configures an interface type and places you in interface configuration mode
no shutdown	Administratively activates an interface
ip address *ip-address mask* [**secondary**]	Sets a primary or secondary IP address for an interface
encapsulation frame-relay [**ietf**]	Enables and specifies the Frame Relay encapsulation method
frame-relay interface-dlci dlci [**protocol ip** ip-address]	Assigns a data link connection identifier (DLCI) to a specified Frame Relay subinterface on the router or access server
show interfaces [**interface-name**]	Displays the statistical information specific to an interface
router ospf *process-id*	Enables OSPF routing, which places you in router configuration mode
network *ip-address wildcard-mask* **area** *area-id*	Defines an interface on which OSPF runs, and defines the area ID for that interface
ip ospf network {**broadcast** | **non-broadcast** | {**point-to-multipoint** [**non-broadcast**]}}	Configures the OSPF network type for a specified interface
ip host *name* [**tcp-port-number**] **address1** [**address2...address8**]	Eases configuration of the lab by defining a static host-name-to-address mapping in the host cache—a suggestion would be to map to the loopback interface with the highest IP address of each router
no ip domain-lookup	Eases configuration of the lab by disabling the IP Domain Name System (DNS)-based host-name-to-address translation

Job Aids

These job aids are available to help you complete the lab exercise(s):

- The laboratory is organized as a number of workgroups that are connected to two common backbones:

    — ISP exchange point, also called the "provider backbone," where two upstream service providers are located

    — Client ISP backbone, where your customer is connected

- Every workgroup has four routers named WG*x*R1, WG*x*R2, WG*x*R3, and WG*x*R4, where *x* is the number of the workgroup. There are also three shared routers called "Good," "Cheap," and "Client."

- You will perform initial router configuration and prepare the routers for further exercises. During this procedure, configure passwords, serial interfaces, and IP addresses on each router in your student workgroup. You will also configure an IGP in your workgroup.

- Figure 1 displays the physical connectivity within your student workgroup. You have control over routers WG*x*R1 through WG*x*R4. You can also Telnet to other routers that are shown in the figure, but you cannot configure them.
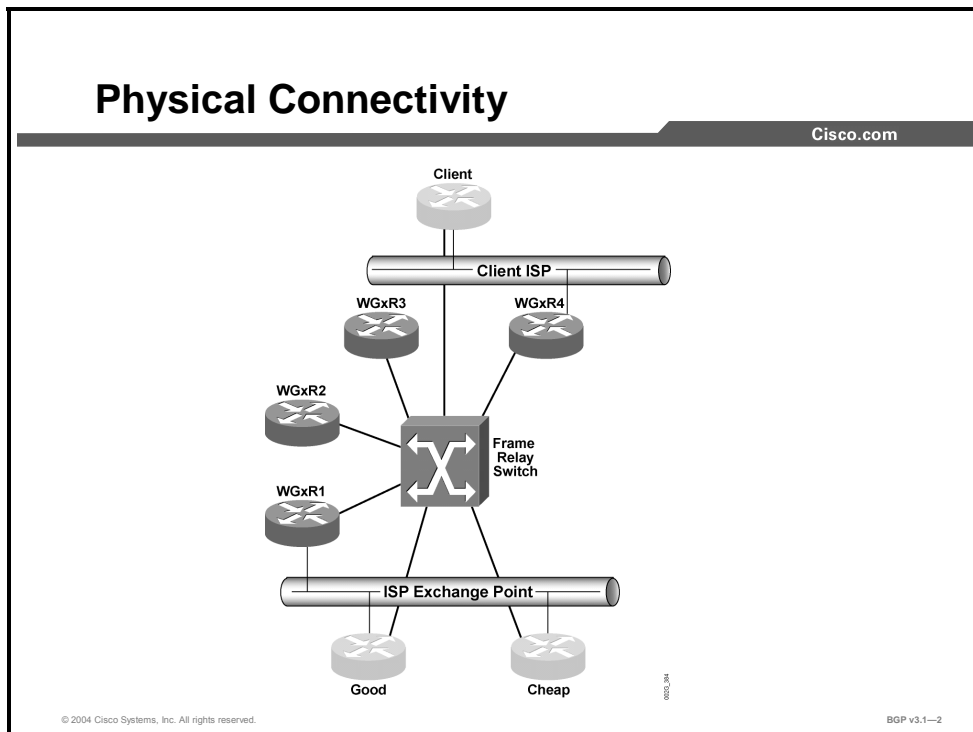
Figure 1: Physical Connectivity

- The first serial interface of each of your routers is connected to the Frame Relay switch. The first (fast) Ethernet interface of each router is connected to the LAN segment. All routers including the shared ones (Good, Cheap, and Client) have one serial link to the Frame Relay switch.

- Figure 2 displays the logical connectivity of your student workgroup. Frame Relay DLCIs are already configured on the Frame Relay switch to provide this topology.
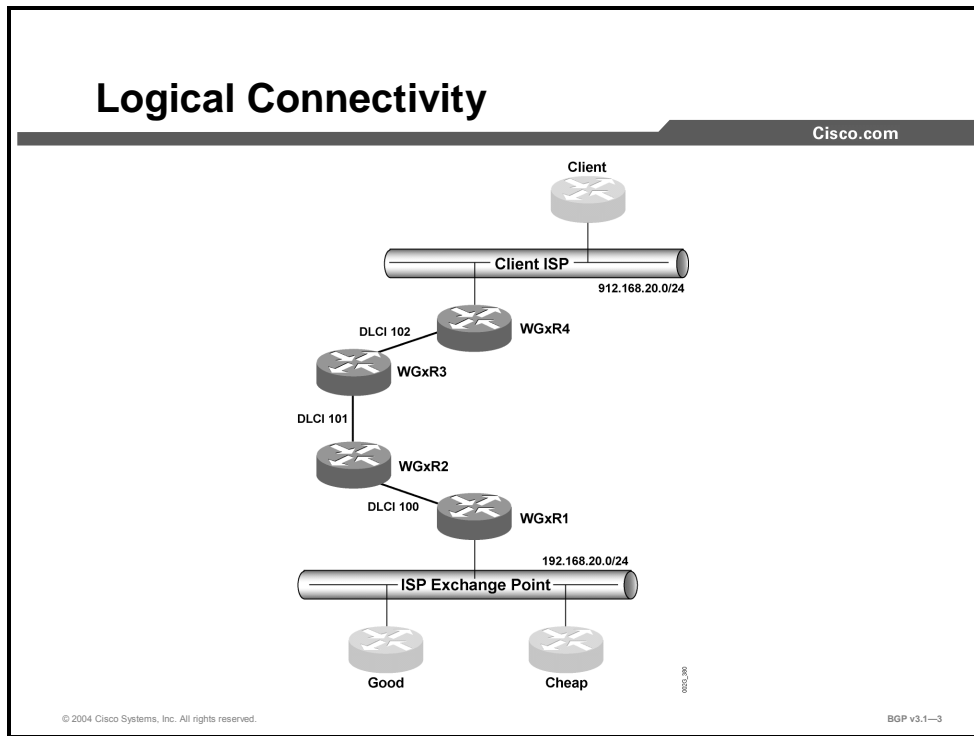
Figure 2: Logical Connectivity

Exercise Procedure

Complete these steps:

Step 1   Perform initial configuration of your routers using the parameters in the following table.

Parameter	Value
host name	Use host names as shown in Figure 2 (*x* is the number of your workgroup).
Enable password	Cisco
VTY password	Cisco
WAN link encapsulation	Frame Relay
WAN link clock rate	128 kbps (configured on the Frame Relay switch)

Step 2 Configure two loopback addresses on each of your workgroup routers with the IP addresses from the following table.

Router	Interface	Address	Subnet Mask
WG*x*R1	Loopback 0	197.*x*.1.1	255.255.255.0
	Loopback 1	197.*x*.8.1	255.255.255.0
WG*x*R2	Loopback 0	197.*x*.2.1	255.255.255.0
	Loopback 1	197.*x*.3.1	255.255.255.0
WG*x*R3	Loopback 0	197.*x*.4.1	255.255.255.0
	Loopback 1	197.*x*.5.1	255.255.255.0
WG*x*R4	Loopback 0	197.*x*.6.1	255.255.255.0
	Loopback 1	197.*x*.7.1	255.255.255.0

Step 3 Configure LAN IP addresses on WG*x*R1 and WG*x*R4 using parameters from the following table.

Parameter	Value
ISP exchange point subnet	192.168.20.*x*, subnet mask 255.255.255.0
Client ISP subnet	192.168.21.*x*, subnet mask 255.255.255.0

Note Router "Good" has IP address 192.168.20.20, and router "Cheap" has IP address 192.168.20.22. They are shared by all workgroups. Router "Client" has IP address 192.168.21.99 and is shared by all workgroups. Frame Relay DLCIs have the same value on both ends of the link.

Step 4 Configure point-to-point Frame Relay subinterfaces on the Frame Relay links. The IP addresses to be used on the link, as well as the DLCI values for the Frame Relay virtual circuits, are shown in the following table.

First Router	IP Address	Second Router	IP Address	DLCI
WG*x*R1	192.168.*x*.1/30	WG*x*R2	192.168.*x*.2/30	100
WG*x*R2	192.168.*x*.5/30	WG*x*R3	192.168.*x*.6/30	101
WG*x*R3	192.168.*x*.9/30	WG*x*R4	192.168.*x*.10/30	102

Step 5 Configure IP host mappings to ease Telnet hopping between routers.

Step 6 Configure any IGP between your routers. Make sure that you do *not* use the IGP on the backbone LANs.

Note It is preferred that you use a classless IGP, such as OSPF, for this step.

Exercise Verification

You have completed this exercise when you attain these results:

- All router interfaces should be active (line up, line protocol up).

- You should be able to Telnet and ping between all core routers.

Lab Exercise 1-2: Configuring Basic BGP

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP. After completing this exercise, you will be able to meet these objectives:

- Configure initial BGP setup
- Configure BGP neighbors
- Announce local networks in BGP
- Redistribute routes into BGP
- Configure basic BGP route aggregation
- Monitor the status of the BGP routing process
- Monitor BGP neighbors
- Monitor the BGP table

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.
- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.
- Two Ethernet 10BASE-T patch cables.
- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed
- Two Cisco 2610 routers with BGP-capable operating system software installed
- One Cisco 3640 router with an installed NM-8A/S
- Two Catalyst 2924M-XL Ethernet switches
- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **remote-as** *as-number*	Establishes a BGP session by using your workgroup number as the AS number
network *network* [**mask** *mask*]	Announces IP prefixes in BGP
ip route *network mask* …	Configures a static IP route
route-map *name* {**permit | deny**} **seq**	Defines or modifies an existing entry in a route-map
match ip address *acl*	Matches routes in a route-map
set origin igp	Sets the origin in a route-map
redistribute *igp pid* **route-map** *name*	Redistributes from your IGP into BGP
aggregate-address *network mask* [**summary-only**]	Creates summary prefixes—use the **summary-only** keyword to suppress more specific prefixes
show ip bgp summary	Verifies if the BGP session is up
show ip bgp neighbor	Enables you to view detailed information about the neighbor
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp *network*	Enables you to view detailed information about prefixes (aggregates)
default-information originate [**always**]	Generates a default external route into an OSPF routing domain—to enable advertising of the default route regardless of whether the software has a default route, use the option **always**

Job Aids

These job aids are available to help you complete the lab exercise:

- You must connect to the Internet using BGP, ensuring that all users in your network have Internet access. You will connect to a single service provider and statically announce the address space that the Internet Registry has assigned to you.

- Figure 1 displays the BGP session that you will establish between WGxR1 and the "Good" service provider.
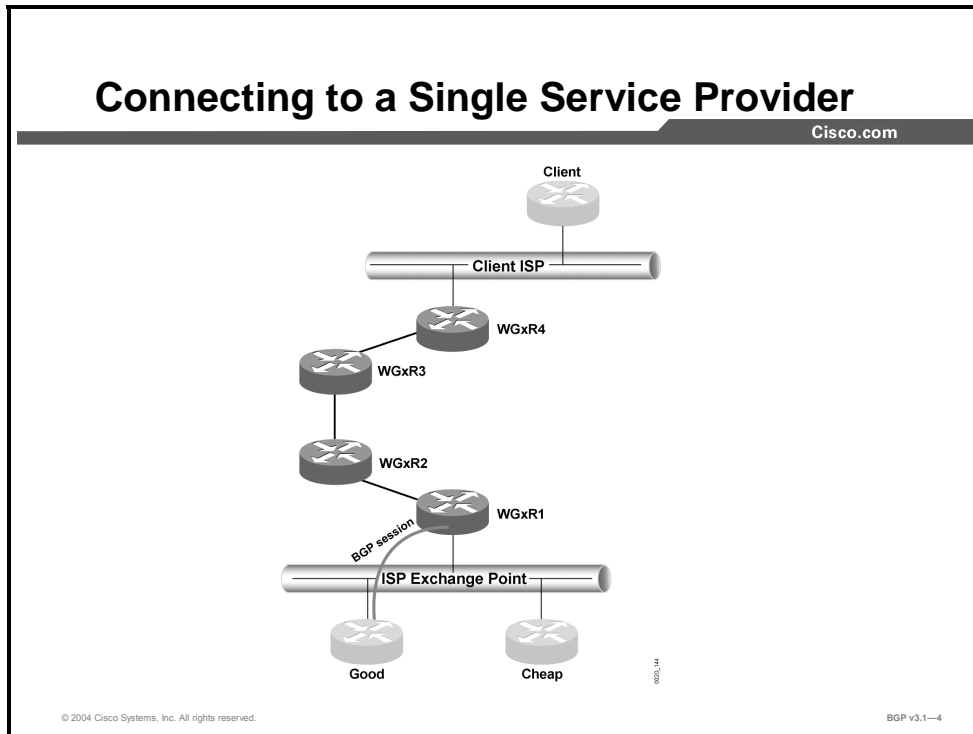
Figure 1: Connecting to a Single Service Provider

Task 1: Configuring BGP

In this task, you will configure your network backbone for basic BGP connectivity with a service provider to establish BGP peering.

Exercise Procedure

Complete these steps:

Step 1   Start the BGP process on WG*x*R1. Use your workgroup number as the AS number.

Step 2   Configure the "Good" router to be your BGP neighbor using the following parameters.

Parameter	Value
Service provider IP address	192.168.20.20
Service provider AS number	20

Step 3   Announce the IP prefix 197.*x*.0.0/16 by configuring it in the BGP routing process. Also announce network 192.168.*x*.0/24.

Step 4   Ensure that WG*x*R1 is the exit point for your pod and is originating a default route for your IGP.

Exercise Verification

You have completed this exercise when you attain these results:

- Verify that you have established a BGP session. Remember that it may take up to a minute to establish a BGP session. Your display should resemble the following:

```
WG1R1#show ip bgp summary
BGP router identifier 192.168.20.1, local AS number 1
BGP table version is 25, main routing table version 25
24 network entries and 24 paths using 3192 bytes of memory
5 BGP path attribute entries using 260 bytes of memory
4 BGP AS-PATH entries using 96 bytes of memory
1 BGP community entries using 250 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 activity 24/0 prefixes, 24/0 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxR
192.168.20.20   4    20       9       4       25    0    0 00:01:31         24
WG1R1#
```

- View detailed information about the neighbor. Your display should resemble the following:

```
WG1R1#show ip bgp neighbor 192.168.20.20
BGP neighbor is 192.168.20.20,  remote AS 20, external link
  BGP version 4, remote router ID 199.199.199.199
  BGP state = Established, up for 00:19:50
  Last read 00:00:50, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received
    Address family IPv4 Unicast: advertised and received
  Received 27 messages, 0 notifications, 0 in queue
  Sent 22 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
```

```
  BGP table version 25, neighbor version 25
  Index 1, Offset 0, Mask 0x2
  24 accepted prefixes consume 864 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0

  Connections established 1; dropped 0
  Last reset never
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 192.168.20.1, Local port: 179
Foreign host: 192.168.20.20, Foreign port: 18395

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
… rest deleted …
```

- Inspect the contents of the BGP table on your router. You should see a large number of networks being advertised by the "Good" provider. Your display should resemble the following:

```
WG1R1#show ip bgp
BGP table version is 32, local router ID is 192.168.20.1
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
*> 128.20.0.0       192.168.20.20            0             0 20 i
*> 128.22.0.0       192.168.20.20            0             0 20 i
*> 128.26.0.0       192.168.20.20            0             0 20 42 26 i
*> 128.37.0.0       192.168.20.20            0             0 20 42 37 i
*> 128.42.0.0       192.168.20.20            0             0 20 42 i
*> 128.51.0.0       192.168.20.20            0             0 20 42 26 51 i
*> 128.213.0.0      192.168.20.20            0             0 20 213 i
*> 128.214.0.0      192.168.20.22                          0 20 22 214 i
*> 192.20.11.0      192.168.20.20            0             0 20 i
*> 192.22.11.0      192.168.20.20            0             0 20 i
*> 192.26.11.0      192.168.20.20            0             0 20 42 26 i
*> 192.37.11.0      192.168.20.20            0             0 20 42 37 i
*> 192.42.11.0      192.168.20.20            0             0 20 42 i
*> 192.51.11.0      192.168.20.20            0             0 20 42 26 51 i
*> 192.168.1.0      0.0.0.0                  0         32768 i
*> 192.168.2.0      192.168.20.2                          0 20 2 i
*> 192.168.3.2/32   192.168.20.3                          0 20 3 ?
*> 192.168.3.3/32   192.168.20.3                          0 20 3 ?
*> 192.168.3.4/32   192.168.20.3                          0 20 3 ?
*> 192.168.3.8/30   192.168.20.3                          0 20 3 ?
*> 192.168.3.12/30  192.168.20.3                          0 20 3 ?
*> 192.168.3.16/30  192.168.20.3                          0 20 3 ?
*> 192.168.3.20/30  192.168.20.3                          0 20 3 ?
*> 192.213.11.0     192.168.20.20            0             0 20 213 i
*> 192.214.11.0     192.168.20.22                          0 20 22 214 i
*> 197.1.0.0/16     0.0.0.0                  0         32768 i
*> 197.2.0.0/16     192.168.20.2                          0 20 2 i
WG1R1#
```

> Networks originating in AS 3. Origin code is "incomplete" because networks are redistributed.

> Network originating in AS 2.

- Telnet from WG*x*R1 into the router "Good" and verify that it is receiving your networks over BGP.

- Verify that you are receiving networks that are announced by other customers.

```
*> 192.168.1.0      192.168.20.1             0             0 1 i
*> 197.1.0.0/16     192.168.20.1             0             0 1 I
```

- Perform ping and trace from WG*x*R4 to 192.20.11.1 (an Internet destination that is announced by router "Good").

Answer these questions:

Q1)   What do you need in order to propagate classful networks?

Q2)   What do you need in order to propagate classless networks (supernets or subnets)?

Q3)   Why do some networks that are received from router "Good" have a next-hop address pointing to other routers?

Q4)   What command would you use to see if a neighbor is sending you any updates and how many?

Task 2: Configuring Route Redistribution in BGP

Your network has grown, and you can no longer rely on manually configuring your address space in the BGP process. In this task, you will use redistribution to announce a large number of networks into the BGP routing process. You will also use route-maps to set the origin of BGP routes to "IGP" instead of "incomplete."

In this task, you will remove all networks from your BGP definitions (from Task 1) and announce them by using redistribution from your IGP into BGP with a route-map, which sets the origin code to "IGP." Make sure that you do not announce 192.168.20.0/24 and 192.168.21.0/24 networks into BGP.

Exercise Procedure

Complete these steps:

Step 1   Remove all BGP network statements from the previous exercise.

Step 2   Telnet to router "Good" and verify that it no longer receives your networks.

Step 3   Configure an access-list that permits all your networks except those that are shared among workgroups.

Step 4   Configure a route-map. Use the new access-list with a **match** command in the route-map. Use the **set** command in the route-map to set the origin to "IGP."

Note   Route-maps will be covered in detail in the module "Route Selection Using Policy Controls."

Step 5   Configure redistribution from your IGP into BGP by using the previously configured route-map.

Exercise Verification

You have completed this exercise when you attain these results:

■ Log into the service provider router ("Good") and verify that it receives proper networks from you. Your display should resemble the following:

```
Good>show ip bgp
BGP table version is 70, local router ID is 199.199.199.199
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
. . .
*> 192.168.1.0/30   192.168.20.1           0            0 1 i
*> 192.168.1.0      192.168.20.1           0            0 1 i
*> 192.168.1.4/30   192.168.20.1     2681856            0 1 i
*> 192.168.1.8/30   192.168.20.1     3193856            0 1 i
*> 197.1.1.0        192.168.20.1           0            0 1 i
*> 197.1.2.0        192.168.20.1     2297856            0 1 i
*> 197.1.3.0        192.168.20.1     2297856            0 1 i
*> 197.1.4.0        192.168.20.1     2809856            0 1 i
*> 197.1.5.0        192.168.20.1     2809856            0 1 i
*> 197.1.6.0        192.168.20.1     3321856            0 1 i
*> 197.1.7.0        192.168.20.1     3321856            0 1 i
*> 197.1.8.0        192.168.20.1           0            0 1 i
. . .
Good>
```

Individual subnets are seen if "**no auto-summary**" command is used in BGP.

■ Make sure that you are not originating networks 192.168.20.0/24 and 192.168.21.0/24.

■ Verify that your networks are removed from the BGP table when they become unavailable (try shutting down one of the loopback interfaces).

Answer these questions:

Q1) What is the major difference between this implementation and the previous one? Which is better and why?

Q2) What precautions do you have to take when using redistribution?

Task 3: Configuring BGP Aggregation

Your ISP has requested that you provide only summarized prefixes for your address range 197.*x*.0.0. However, because of diagnostic needs, you still need to announce the network 197.*x*.8.0.

In this task, you will configure BGP aggregation as requested, using the **aggregate-address** command.

Exercise Procedure

Complete these steps:

Step 1   Log into the router "Good"; verify that the ISP sees all your individual loopback networks.

```
> 197.1.1.0          192.168.20.1             0              0 1 i
> 197.1.2.0          192.168.20.1             0              0 1 i
> 197.1.3.0          192.168.20.1             0              0 1 i
> 197.1.4.0          192.168.20.1             0              0 1 i
> 197.1.5.0          192.168.20.1             0              0 1 i
> 197.1.6.0          192.168.20.1             0              0 1 i
> 197.1.7.0          192.168.20.1             0              0 1 i
> 197.1.8.0          192.168.20.1             0              0 1 i
```

Step 2   Use the **aggregate** command in your BGP process to aggregate the 197.*x*.0.0 network as requested by your ISP.

Note   Use the keyword **summary-only** if you do not want to announce individual prefixes.

Exercise Verification

You have completed this exercise when you attain these results:

■   Check the BGP table on your router WG*x*R1 to verify the correct routing information is present.

■   Log into the service provider ("Good") router and check its BGP table. Suppressed networks should not be visible on router "Good."

■   View detailed information about one of the aggregates.

Answer these questions:

Q1)   Do you see all your prefixes on the provider router? Why?

Q2)   What do you need to be able to generate and propagate aggregate routes?

Lab Exercise 2-1: Configuring a Transit AS

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will enable a service provider network with multiple BGP connections to other autonomous systems to behave as a transit AS. After completing this exercise, you will be able to meet these objectives:

- Plan the migration of an existing backbone toward a fully meshed IBGP backbone that is designed for transit traffic

- Configure IBGP sessions between loopback interfaces

- Configure BGP synchronization to ensure successful IBGP operation of the transit AS

- Monitor IBGP operation

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **remote-as** *as-number*	Establishes an IBGP session by using your workgroup number as the AS number
neighbor *ip-address* **update-source** *interface*	Uses the IP address of the specified interface as the source address for the BGP session
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
neighbor *ip-address* **next-hop-self**	Configures the router as the next hop for a BGP-speaking neighbor
no synchronization	Disables synchronization of IGP and BGP routes

Job Aids

These job aids are available to help you complete the lab exercise:

■ With the rapid growth of the Internet, you decide to become an Internet service provider (ISP), and you already have your first customer.

■ In this exercise, you will transform your network into a transit AS that runs BGP on all core routers.

■ Configure a full mesh of IBGP sessions between all routers in your AS. Establish these BGP sessions between Loopback 0 interfaces.

■ Figure 1 displays the required BGP connectivity within your AS as well as the BGP sessions with your customer and your upstream ISPs.
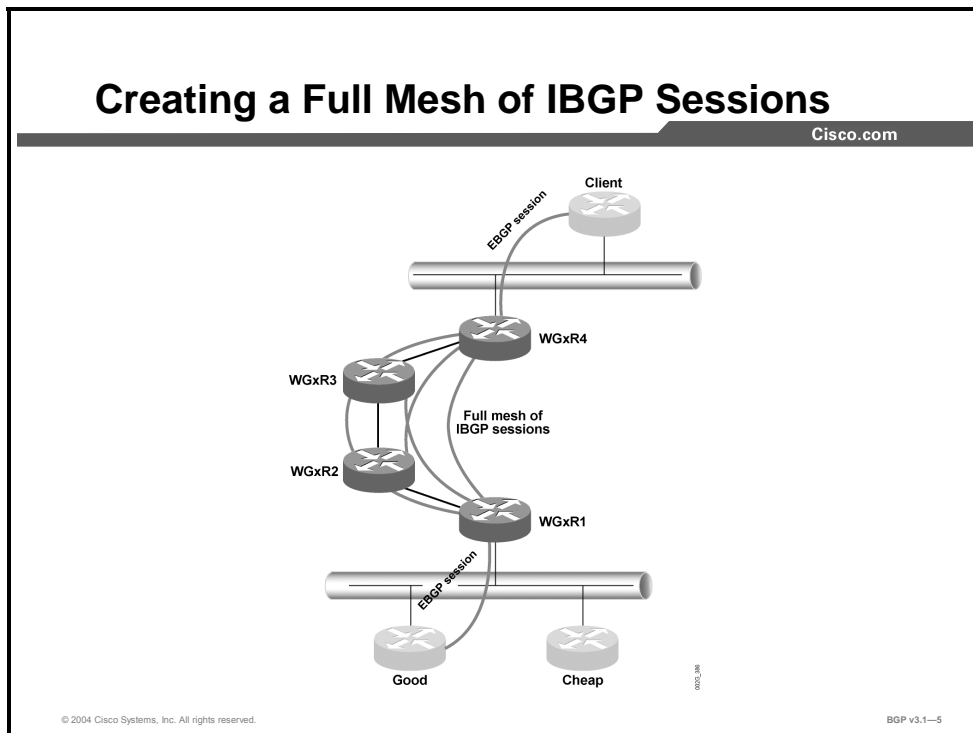
Figure 1: Creating a Full Mesh of IBGP Sessions

Task 1: Configuring the BGP Transit AS

In this task, you will configure your network backbone as a fully meshed IBGP backbone acting as a transit AS.

Exercise Procedure

Complete these steps:

Step 1 Clean up your BGP configuration by removing the BGP process and the default route from the IGP on WG*x*R1. You may also remove the access-list and route-map.

Step 2 Start the BGP process on all routers in your workgroup.

Step 3 Advertise your prefixes (197.*x*.0.0/16 and 192.168.*x*.0/24) in BGP on routers WG*x*R1 and WG*x*R4.

Step 4 Re-establish the neighbor relationships to router "Good" without any filters on router WG*x*R1. Use the parameters from the following table.

Router	AS Number	IP Address
Good	20	192.168.20.20

Step 5 Establish a BGP session with router "Client" on router WG*x*R4. Use the parameters from the following table.

Parameter	Value
Client IP address	192.168.21.99
Client AS number	99

Step 6 Configure all routers in your workgroup as IBGP neighbors (IBGP full mesh). Use loopback interfaces to establish these IBGP sessions.

Step 7 Ensure EBGP reachability by all IBGP-speaking routers within your transit AS without redistributing the connected interfaces of either external Ethernet.

Exercise Verification

You have completed this exercise when you attain these results:

- Check BGP on all core routers and the router "Client" and ensure that they have established the correct sessions with their peers. Your display should resemble the following:

```
WG1R1#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.20.20   4    20   1189    1200       52    0    0 13:34:45       23
192.168.20.22   4    22   1195    1183       52    0    0 13:34:46       23
197.1.2.1       4     1   1174    1196       52    0    0 13:34:58        0
197.1.4.1       4     1   1170    1188       52    0    0 13:35:17        0
197.1.6.1       4     1   1171    1183       52    0    0 13:34:58       13


WG1R2#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
197.1.1.1       4     1   1200    1179      125    0    0 13:38:09       24
197.1.4.1       4     1   1173    1173      125    0    0 13:38:31        0
197.1.6.1       4     1   1176    1170      125    0    0 13:37:58       13
```

```
WG1R3#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
197.1.1.1       4     1    1193    1175       78    0    0 13:40:33        24
197.1.2.1       4     1    1175    1175       78    0    0 13:40:37         0
197.1.6.1       4     1    1183    1175       78    0    0 13:40:04        13


WG1R4#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.21.99   4    99    1191    1192       14    0    0 13:40:38        11
197.1.1.1       4     1    1190    1178       14    0    0 13:41:04        24
197.1.2.1       4     1    1173    1179       14    0    0 13:40:54         0
197.1.4.1       4     1    1175    1183       14    0    0 13:40:54         0


Client#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.21.X    4     1    1147    1146       18    0    0 13:42:01         2
```

■ Check the BGP table on router "Client" and verify that it is correctly receiving BGP routes.
  Your display should resemble the following:

```
Client#show ip bgp
BGP table version is 119, local router ID is 197.99.111.1
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
*> 10.0.0.0         192.168.21.1                            0 1 20 i
*> 99.0.0.0         0.0.0.0                  0          32768 i
*> 128.20.12.0/24   192.168.21.1                            0 1 20 i
*> 128.37.0.0       192.168.21.1                            0 1 20 42 37 i
*> 128.42.0.0       192.168.21.1                            0 1 20 42 i
*> 128.213.0.0      192.168.21.1                            0 1 20 213 i
*> 192.20.12.0/30   192.168.21.1                            0 1 20 i
*> 192.37.11.0      192.168.21.1                            0 1 20 42 37 i
*> 192.42.11.0      192.168.21.1                            0 1 20 42 i
*> 192.168.1.0      192.168.21.1             0              0 1 i
*> 192.213.11.0     192.168.21.1                            0 1 20 213 i
*> 197.1.0.0/16     192.168.21.1             0              0 1 i
*> 197.99.1.0       0.0.0.0                  0          32768 i
*> 197.99.11.0      0.0.0.0                  0          32768 i
*> 197.99.12.0      0.0.0.0                  0          32768 i
*> 197.99.13.0      0.0.0.0                  0          32768 i
*> 197.99.20.0      0.0.0.0                  0          32768 i
*> 197.99.22.0      0.0.0.0                  0          32768 i
*> 197.99.80.0      0.0.0.0                  0          32768 i
*> 197.99.111.1/32  0.0.0.0                  0          32768 i
*> 197.99.120.0     0.0.0.0                  0          32768 i
*> 197.99.128.0/20  0.0.0.0                  0          32768 i
*> 200.20.0.0/16    192.168.21.1                            0 1 20 i
```

■ Use traceroute from router WG*x*R1 to the loopback interface on router "Client"
  (197.99.1.1). You should see a path similar to the one here:

```
WG1R1# traceroute 197.99.1.1
Type escape sequence to abort.
Tracing the route to 197.99.1.1

  1 192.168.1.2 16 msec 16 msec 17 msec
  2 192.168.1.6 32 msec 32 msec 28 msec
  3 192.168.1.10 44 msec 40 msec 40 msec
  4 192.168.21.99 48 msec *  44 msec
```

Answer these questions:

Q1)   Check the BGP table on router "Client." How many prefixes coming from your AS are in that BGP table? _____

Q2)   Is there any other way of discovering how many prefixes that you have advertised to the router "Client"?

Q3)   Why did you have to disable synchronization?

Q4)   Why did you have to establish a full mesh of IBGP sessions?

Lab Exercise 3-1: Using Multihomed BGP Networks

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP as part of a customer scenario where you must support connections to multiple ISPs. After completing this exercise, you will be able to meet these objectives:

- Configure BGP neighbors to support a multihomed customer scenario
- Monitor the status of the BGP routing process
- Monitor BGP neighbors in a multihomed customer scenario

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.
- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.
- Two Ethernet 10BASE-T patch cables.
- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed
- Two Cisco 2610 routers with BGP-capable operating system software installed
- One Cisco 3640 router with an installed NM-8A/S
- Two Catalyst 2924M-XL Ethernet switches
- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **remote-as as**	Starts a BGP session with the neighboring AS
neighbor ip-address weight *weight*	Assigns a weight to all updates that are received from the specified neighbor
show ip bgp summary	Verifies the state of BGP sessions
show ip bgp	Enables you to inspect the contents of the BGP table

Job Aids

These job aids are available to help you complete the lab exercise:

■ You have started to provide mission-critical e-commerce services, and you must ensure their high availability. You decide to connect to a new ISP, "Cheap," using "Cheap" as your primary ISP and "Good" as your backup ISP.

■ Figure 1 shows the connectivity that you need to establish for the second BGP session with the new provider router ("Cheap").
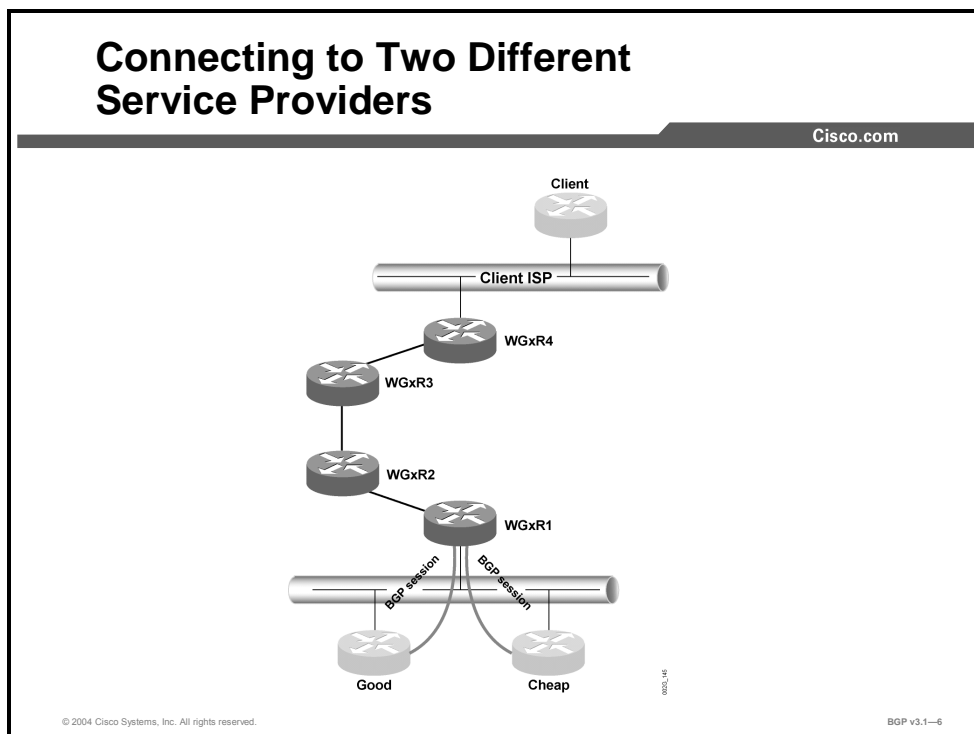

Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Step 1 Remove BGP from routers WG*x*R2, WG*x*R3, and WG*x*R4. Leave only the existing EBGP session on router WG*x*R1 to "Good." Because only WG*x*R1 is running BGP, enable synchronization and restore IGP default routing for your pod.

Step 2 Configure the second BGP neighbor on WG*x*R1 using the parameters in the following table.

Service Provider	Parameter	Value
Cheap	AS number	22
	IP address	192.168.20.22

Step 3 Change the default weight on neighbor "Cheap" to 100 using the **neighbor** *ip-address* **weight** command to give preference to routes that are announced from router "Cheap" over those announced from router "Good."

Note BGP weights are covered in detail in the module "Route Selection Using Attributes" and its supporting lab exercises.

Exercise Verification

You have completed this exercise when you attain these results:

■ Verify that both BGP sessions are up. Your display should resemble the following:

```
WG1R1#show ip bgp summary
BGP router identifier 197.1.8.1, local AS number 1
BGP table version is 122, main routing table version 122
35 network entries and 58 paths using 5483 bytes of memory
18 BGP path attribute entries using 936 bytes of memory
16 BGP AS-PATH entries using 384 bytes of memory
0 BGP community entries using 0 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 activity 49/464 prefixes, 95/37 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down  State/PfxR
192.168.20.20   4    20     411     412       122    0    0 00:07:10         23
192.168.20.22   4    22      45      34       122    0    0 00:07:11         23
WG1R1#
```

> Both sessions are up, and 23 prefixes were received from both neighbors.

■ Check all the BGP prefixes with two paths to verify that your router prefers the one through router "Cheap." Your display should resemble the following:

```
wg1r1#show ip bgp
BGP table version is 122, local router ID is 197.1.8.1
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
*> 10.0.0.0         192.168.20.233                       100 22 20 i
*                   192.168.20.233           0             0 20 i
```

```
*> 128.20.0.0        192.168.20.22        0            100 22 i
*                    192.168.20.20        0              0 20 i
*> 128.20.12.0/24    192.168.20.20                     100 22 20 i
*                    192.168.20.20        0              0 20 i
*> 128.22.0.0        192.168.20.22        0            100 22 i
*                    192.168.20.20        0              0 20 i
*> 128.22.12.0/24    192.168.20.22        0            100 22 i
*                    192.168.20.22                       0 20 22 i
*> 128.26.0.0        192.168.20.22        0            100 22 26 i
*                    192.168.20.20        0              0 20 42 26 i
*> 128.37.0.0        192.168.20.22        0            100 22 26 42 37 i
*                    192.168.20.20        0              0 20 42 37 i
*> 128.42.0.0        192.168.20.22        0            100 22 26 42 i
*                    192.168.20.20        0              0 20 42 i
*> 128.51.0.0        192.168.20.22        0            100 22 26 51 i
*                    192.168.20.20        0              0 20 42 26 51 i
*> 128.213.0.0       192.168.20.20                     100 22 20 213 i
*                    192.168.20.20        0              0 20 213 i
*> 128.214.0.0       192.168.20.22        0            100 22 214 i
*                    192.168.20.22                       0 20 22 214 i
*> 192.20.11.0       192.168.20.22        0            100 22 i
*                    192.168.20.20        0              0 20 i
*> 192.20.12.0/30    192.168.20.20                     100 22 20 i
*                    192.168.20.20        0              0 20 i
*> 192.22.11.0       192.168.20.22        0            100 22 i
*                    192.168.20.20        0              0 20 i
*> 192.22.12.0/30    192.168.20.22        0            100 22 i
*                    192.168.20.22                       0 20 22 i
*> 192.26.11.0       192.168.20.22        0            100 22 26 i
*                    192.168.20.20        0              0 20 42 26 i
*> 192.37.11.0       192.168.20.22        0            100 22 26 42 37 i
*                    192.168.20.20        0              0 20 42 37 i
*> 192.42.11.0       192.168.20.22        0            100 22 26 42 i
*                    192.168.20.20        0              0 20 42 i
*> 192.51.11.0       192.168.20.22        0            100 22 26 51 i
*                    192.168.20.20        0              0 20 42 26 51 i
```

Answer these questions:

Q1) What can happen if a multihomed AS is passing routing information, learned from one neighbor, to another neighbor?

Q2) Why do some prefixes have two paths but both use the same next-hop address?

Lab Exercise 3-2: Employing AS-Path Filters

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection using AS-path filters where connections to multiple ISPs must be supported. After completing this exercise, you will be able to meet these objectives:

- Identify where you can apply an AS-path filter when configuring a router to influence route selection

- Configure AS-path filters to influence route selection

- Monitor the operation of configured AS-path filters

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **filter-list** *as-path-filter* {**in** | **out**}	Filters incoming or outgoing updates with the specified AS-path filter
ip as-path access-list *number* **permit** *regexp*	Specifies the AS-path filter
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
show ip bgp filter-list *access-list-number*	Enables you to check a filter
clear ip bgp *neighbor*	Clears the BGP session

Job Aids

These job aids are available to help you complete the lab exercise:

- Your service providers complain that you propagate routes between them and that you could potentially become a transit network.

- You have also discovered that the service provider "Cheap" does not provide you with good connectivity to your users in AS 213. Therefore, you want to accept the routes originating in AS 213 only from the provider "Good."

- Customers in AS 214 request that you use only the provider "Cheap" to communicate with them because of security reasons.

- Use AS-path filters to filter incoming routes and to limit propagation of transit routes.

- Select routes between the "Good" and "Cheap" service providers based on the following criteria:

  — Accept routes originating in AS 213 only from router "Good."

  — Accept routes with AS 214 in the AS path only from router "Cheap."

  — Ensure that you do not act as a transit AS between the two service provider routers ("Good" and "Cheap").

- Figure 1 shows the structure of the autonomous systems beyond AS 20 and AS 22. This information will assist you in your verification and any troubleshooting steps.
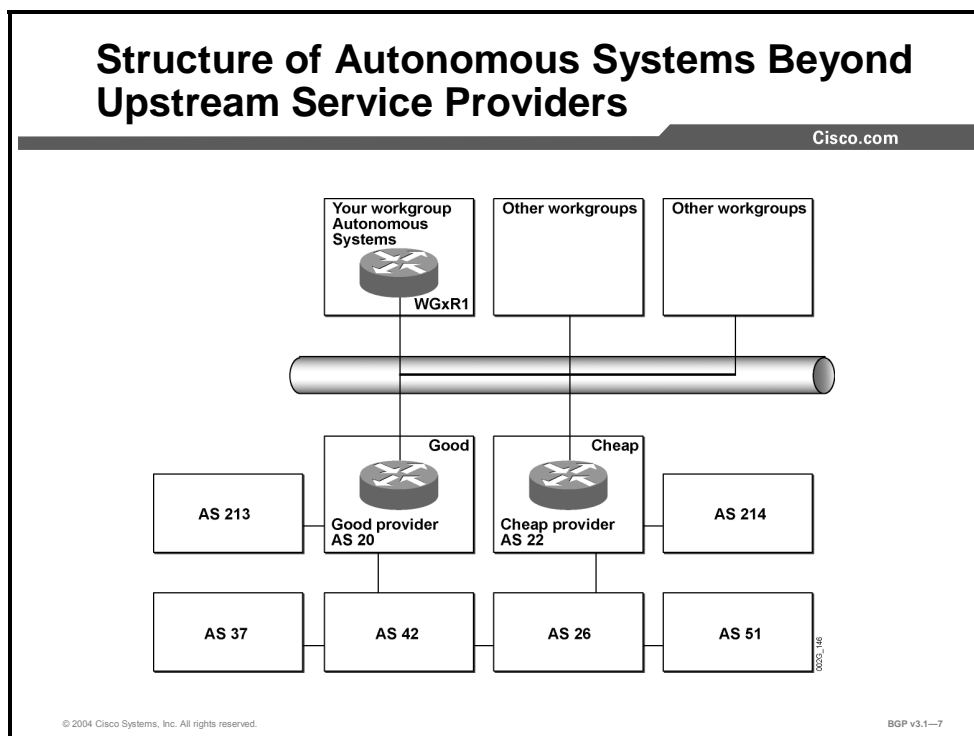
Figure 1: Structure of Autonomous Systems Beyond Your Upstream Service Providers

Exercise Procedure

Complete these steps:

Step 1   Log on to the "Good" and "Cheap" routers and verify the service provider claims that you propagate the routes between them. Your display should resemble the following:

Router "Good":

```
   Network          Next Hop          Metric LocPrf Weight Path
*  192.168.1.0      192.168.20.1                        0 22 1 i
*>                  192.168.20.1          0             0 1 i
*  197.1.0.0/16     192.168.20.1                        0 22 1 i
*>                  192.168.20.1                        0 1 i
```

Router "Cheap":

```
   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.1.0      192.168.20.1          0             0 1 i
*                   192.168.20.1                        0 20 1 i
*> 197.1.0.0/16     192.168.20.1                        0 1 i
*                   192.168.20.1                        0 20 1 i
```

Configuring AS-path access-lists:

Step 2   With the **ip as-path access-list** command, configure an AS-path access-list to deny routes that have number "213" as the last number in the AS path. The filter should permit all other routes.

Step 3   Configure another AS-path access-list to deny routes that contain number 214 as the last number in the AS path. The filter should permit all other routes.

Step 4   Configure an AS-path access-list to permit prefixes originating in your AS and deny all other prefixes.

The BGP table showing prefixes that have "213" or "214" in their path (before applying the AS-path filters) should look like this:

```
wg1r1#show ip bgp
BGP table version is 122, local router ID is 197.1.8.1
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
*> 128.213.0.0      192.168.20.20                    100 22 20 213 i
*                   192.168.20.20         0             0 20 213 i
*> 128.214.0.0      192.168.20.22         0           100 22 214 i
*> 192.213.11.0     192.168.20.20                    100 22 20 213 i
*                   192.168.20.20         0             0 20 213 i
*> 192.214.11.0     192.168.20.22         0           100 22 214 i
*                   192.168.20.22         0             0 22 214 i
```

Applying AS-path access-lists:

Step 5   Apply the AS-path access-lists that you configured in the previous steps to your BGP neighbors.

Exercise Verification

You have completed this exercise when you attain these results:

- Verify that all paths containing AS 213 are received only from router "Good" and that all paths containing AS 214 are received only from router "Cheap." Your display should resemble the following:

```
WG1R1#show ip bgp
BGP table version is 84, local router ID is 197.1.8.1
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
*  128.20.0.0       192.168.20.20          0               20 i
*>                  192.168.20.22          0               22 i
*  128.22.0.0       192.168.20.20          0               20 i
*>                  192.168.20.22          0               22 i
*  128.26.0.0       192.168.20.20          0               20 42 26 i
*>                  192.168.20.22          0               22 26 i
*  128.37.0.0       192.168.20.20          0               20 42 37 i
*>                  192.168.20.22          0               22 26 42 37 i
*  128.42.0.0       192.168.20.20          0               20 42 i
*>                  192.168.20.22          0               22 26 42 i
*  128.51.0.0       192.168.20.20          0               20 42 26 51 i
*>                  192.168.20.22          0               22 26 51 i
*> 128.213.0.0      192.168.20.20          0               20 213 i
*> 128.214.0.0      192.168.20.22          0               22 214 i
*  192.20.11.0      192.168.20.20          0               20 i
*>                  192.168.20.22          0               22 i
*  192.22.11.0      192.168.20.20          0               20 i
*>                  192.168.20.22          0               22 i
*  192.26.11.0      192.168.20.20          0               20 42 26 i
*>                  192.168.20.22          0               22 26 i
*  192.37.11.0      192.168.20.20          0               20 42 37 i
*>                  192.168.20.22          0               22 26 42 37 i
*  192.42.11.0      192.168.20.20          0               20 42 i
*>                  192.168.20.22          0               22 26 42 i
*  192.51.11.0      192.168.20.20          0               20 42 26 51 i
*>                  192.168.20.22          0               22 26 51 i
. . .
*  192.213.11.0     192.168.20.20          0               20 213 i
*> 192.214.11.0     192.168.20.22          0               22 214 I
. . .
WG1R1#
```

- Verify that routers "Good" and "Cheap" are not receiving any prefixes that do not originate in your AS. Your display should resemble the following:

```
Good> show ip bgp reg ^1_
BGP table version is 47, local router ID is 199.199.199.199
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
*> 192.168.1.0      192.168.20.1           0             0 1 i
*> 197.1.0.0/16     192.168.20.1                         0 1 i
```

Answer these questions:

Q1)   By this time, the BGP table should contain a large number of prefixes. What regular expression would you use with the **show ip bgp command** on WG*x*R1 to view the prefixes that are originated by your AS?

Q2)   What regular expression would you use on router "Good" or "Cheap" to view the prefixes that originate in your AS?

Q3)   What regular expression would you use on router "Good" or "Cheap" to view the prefixes that are received from your AS?

Q4)   How can you test your AS-path filters before applying them and clearing BGP neighbors?

Lab Exercise 3-3: Filtering with Prefix-Lists

Complete this lab exercise to practice what you have learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection using prefix-list filters where you must support connections to multiple ISPs. After completing this exercise, you will be able to meet these objectives:

- Configure filtering of inbound or outbound updates with prefix-lists

- Modify configured prefix-list filters

- Monitor the operation of configured prefix-list filters

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
ip prefix-list *name* [**seq** *seq*] {**permit|deny**} *network/length* [**ge** *length* | **le** *length*]	Enables you to create an entry in a prefix-list
neighbor *ip-address* **prefix-list** *prefix-list* **in**	Filters incoming prefixes
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
show ip bgp prefix-list *prefix-list-name*	Displays a BGP routing table based on the contents of the prefix-list

Job Aids

These job aids are available to help you complete the lab exercise:

- The memory consumption on WG*x*R1 has increased because of the large BGP tables that have been received from your service providers. You realize that the service providers announce a large number of very small prefixes to you. Some of the announced prefixes are also in the private IP (RFC 1918) address space.

- In this exercise, you will minimize the size of the BGP table on your router with inbound filters that are implemented with prefix-lists. You will also filter any prefixes in the private IP address space that you might receive from your service providers.

- Your analysis shows that you must perform the following tasks to minimize the size of the BGP table on router WG*x*R1:

    — Do not accept any subnets of class B networks from router "Cheap."

    — Do not accept class C networks or their subnets from router "Good."

    — Do not accept any private networks (RFC 1918) from either neighbor.

    — Use prefix-lists to perform these tasks.

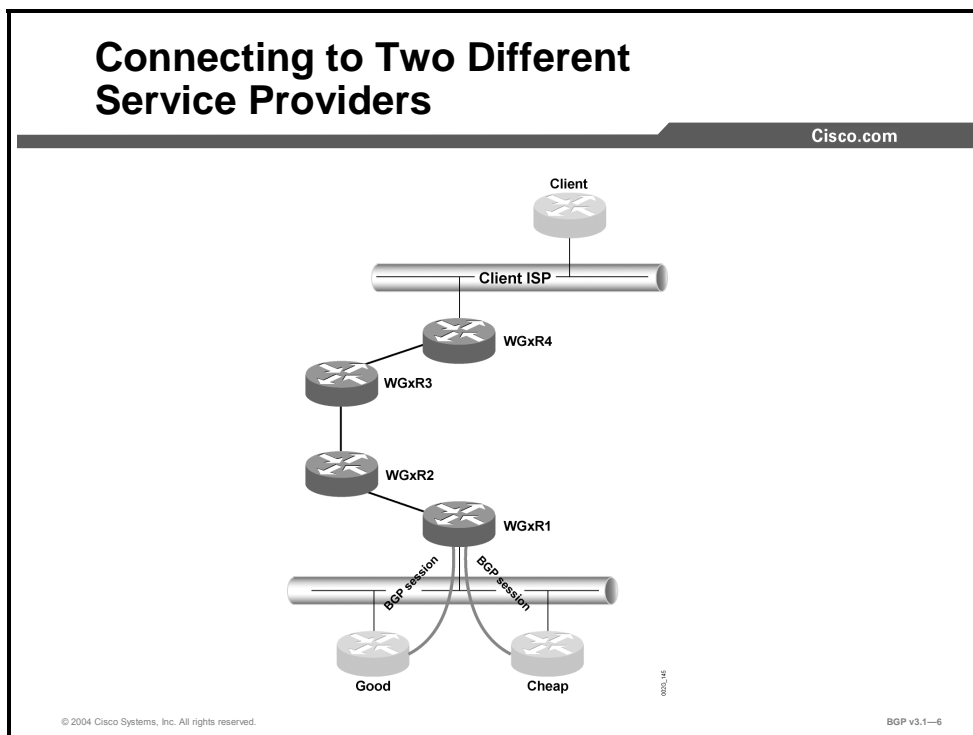- Figure 1 shows the connectivity that is established between your AS and the two service providers "Good" and "Cheap."

Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Step 1   Verify that your current BGP table looks somewhat like the following:

```
wg1r1#show ip bgp
BGP table version is 44, local router ID is 197.1.8.1
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
*  10.0.0.0         192.168.20.233        0             0 20 i
*>                  192.168.20.233                    100 22 20 i
*  128.20.0.0       192.168.20.20         0             0 20 i
*>                  192.168.20.22         0           100 22 i
*  128.20.12.0/24   192.168.20.20         0             0 20 i
*>                  192.168.20.20                      100 22 20 i
*  128.22.0.0       192.168.20.20         0             0 20 i
*>                  192.168.20.22         0           100 22 i
*  128.22.12.0/24   192.168.20.22                       0 20 22 i
*>                  192.168.20.22         0           100 22 i
*  128.26.0.0       192.168.20.20         0             0 20 42 26 i
*>                  192.168.20.22         0           100 22 26 i
*  128.37.0.0       192.168.20.20         0             0 20 42 37 i
*>                  192.168.20.22         0           100 22 26 42 37 i
*  128.42.0.0       192.168.20.20         0             0 20 42 i
*>                  192.168.20.22         0           100 22 26 42 i
*  128.51.0.0       192.168.20.20         0             0 20 42 26 51 i
*>                  192.168.20.22         0           100 22 26 51 i
*> 128.213.0.0      192.168.20.20         0             0 20 213 i
*> 128.214.0.0      192.168.20.22         0           100 22 214 i
*  192.20.11.0      192.168.20.20         0             0 20 i
*>                  192.168.20.22         0           100 22 i
*  192.20.12.0/30   192.168.20.20         0             0 20 i
*>                  192.168.20.20                      100 22 20 i
*  192.22.11.0      192.168.20.20         0             0 20 i
*>                  192.168.20.22         0           100 22 i
*  192.22.12.0/30   192.168.20.22                       0 20 22 i
*>                  192.168.20.22         0           100 22 i
*  192.26.11.0      192.168.20.20         0             0 20 42 26 i
*>                  192.168.20.22         0           100 22 26 i
*  192.37.11.0      192.168.20.20         0             0 20 42 37 i
*>                  192.168.20.22         0           100 22 26 42 37 i
*  192.42.11.0      192.168.20.20         0             0 20 42 i
*>                  192.168.20.22         0           100 22 26 42 i
*  192.51.11.0      192.168.20.20         0             0 20 42 26 51 i
*>                  192.168.20.22         0           100 22 26 51 i
*> 192.168.1.0      0.0.0.0               0         32768 i
*> 192.213.11.0     192.168.20.20         0             0 20 213 i
*> 192.214.11.0     192.168.20.22         0           100 22 214 i
*> 197.1.0.0/16     0.0.0.0                         32768 i
*  200.20.0.0/16    192.168.20.20         0             0 20 i
*>                  192.168.20.20                      100 22 20 i
*  200.22.0.0/16    192.168.20.22                       0 20 22 i
*>                  192.168.20.22         0           100 22 i
```

Configuring prefix-lists:

Step 2   Based on your analysis, create prefix-lists to filter incoming updates from routers "Good" and "Cheap."

Applying prefix-lists:

Step 3   Apply the prefix-lists that you have created to updates coming from routers "Good" and "Cheap."

Exercise Verification

You have completed this exercise when you attain these results:

- Inspect your BGP table and check the proper operation of your prefix-list filters. If your Cisco IOS software version is Release 12.2(11)T or later, you can also issue the **show ip bgp prefix-list** *prefix-list-name* command. Your display should resemble the following:

```
wg1r1#show ip bgp reg ^20_

BGP table version is 50, local router ID is 197.1.8.1
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
*  128.20.0.0       192.168.20.20          0             0 20 i
*> 128.20.12.0/24   192.168.20.20          0             0 20 i
*  128.22.0.0       192.168.20.20          0             0 20 i
*> 128.22.12.0/24   192.168.20.22                        0 20 22 i
*  128.26.0.0       192.168.20.20          0             0 20 42 26 i
*  128.37.0.0       192.168.20.20          0             0 20 42 37 i
*  128.42.0.0       192.168.20.20          0             0 20 42 i
*  128.51.0.0       192.168.20.20          0             0 20 42 26 51 i
*> 128.213.0.0      192.168.20.20          0             0 20 213 i
*  200.20.0.0/16    192.168.20.20          0             0 20 i
*  200.22.0.0/16    192.168.20.22                        0 20 22 i
wg1r1#
```

Full BGP table:

```
wg1r1#show ip bgp
BGP table version is 50, local router ID is 197.1.8.1
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
*> 128.20.0.0       192.168.20.22          0           100 22 i
*                   192.168.20.20          0             0 20 i
*> 128.20.12.0/24   192.168.20.20          0             0 20 i
*> 128.22.0.0       192.168.20.22          0           100 22 i
*                   192.168.20.20          0             0 20 i
*> 128.22.12.0/24   192.168.20.22                        0 20 22 i
*> 128.26.0.0       192.168.20.22          0           100 22 26 i
*                   192.168.20.20          0             0 20 42 26 i
*> 128.37.0.0       192.168.20.22          0           100 22 26 42 37 i
*                   192.168.20.20          0             0 20 42 37 i
*> 128.42.0.0       192.168.20.22          0           100 22 26 42 i
*                   192.168.20.20          0             0 20 42 i
*> 128.51.0.0       192.168.20.22          0           100 22 26 51 i
*                   192.168.20.20          0             0 20 42 26 51 i
*> 128.213.0.0      192.168.20.20          0             0 20 213 i
*> 128.214.0.0      192.168.20.22          0           100 22 214 i
*> 192.20.11.0      192.168.20.22          0           100 22 i
*> 192.20.12.0/30   192.168.20.20                      100 22 20 i
*> 192.22.11.0      192.168.20.22          0           100 22 i
*> 192.22.12.0/30   192.168.20.22          0           100 22 i
*> 192.26.11.0      192.168.20.22          0           100 22 26 i
*> 192.37.11.0      192.168.20.22          0           100 22 26 42 37 i
*> 192.42.11.0      192.168.20.22          0           100 22 26 42 i
*> 192.51.11.0      192.168.20.22          0           100 22 26 51 i
*> 192.168.1.0      0.0.0.0                0         32768 i
*> 192.214.11.0     192.168.20.22          0           100 22 214 i
*> 197.1.0.0/16     0.0.0.0                0         32768 i
*> 200.20.0.0/16    192.168.20.20                      100 22 20 i
*                   192.168.20.20          0             0 20 i
*> 200.22.0.0/16    192.168.20.22          0           100 22 i
*                   192.168.20.22                        0 20 22 i
```

Answer these questions:

Q1)   Are neighboring workgroups still reachable?

Q2)   Why does router WG*x*R1 still accept class B networks from router "Cheap"?

Lab Exercise 3-4: Implementing Changes in BGP Policy

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure the soft reconfiguration feature to minimize the impact of expediting BGP policy updates in a network scenario where you must support connections to multiple ISPs. After completing this exercise, you will be able to meet these objectives:

- Configure and perform a soft reconfiguration
- Monitor the operation of a soft reconfiguration

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.
- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.
- Two Ethernet 10BASE-T patch cables.
- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed
- Two Cisco 2610 routers with BGP-capable operating system software installed
- One Cisco 3640 router with an installed NM-8A/S
- Two Catalyst 2924M-XL Ethernet switches
- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **soft-reconfiguration inbound**	Enables inbound soft reconfiguration
show ip bgp	Enables you to inspect the contents of the BGP table
show ip neighbor *ip-address*	Shows detailed information about a specific neighbor
show ip neighbor *ip-address* **received-routes**	Enables you to view all updates that are received from the specified neighbor (before filters)
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command

Job Aids

These job aids are available to help you complete the lab exercise:

■ Your customers complain that they are sometimes unable to reach your e-commerce servers. You suspect that the frequent changes in your BGP routing policies might be the reason and you would like to implement a mechanism that enables you to perform nondisruptive changes in routing policies. Unfortunately, your service providers run a Cisco IOS release that does not support BGP route refresh functionality.

■ In this exercise, you will configure soft reconfiguration to enable nondisruptive changes in BGP routing policies.

■ Figure 1 displays the required BGP connectivity within your workgroup, as well as the BGP sessions with two different service providers.
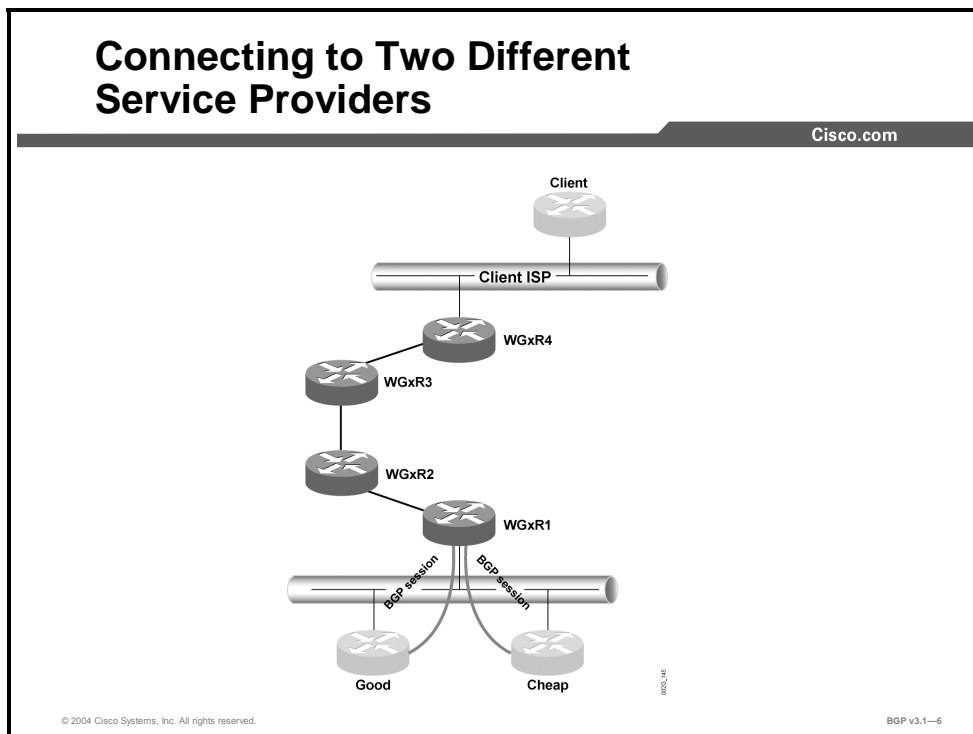

Connecting to Two Different Service Providers

Cisco.com

Client

Client ISP

WGxR4

WGxR3

WGxR2

WGxR1

BGP session   BGP session

Good   Cheap

BGP v3.1—6

Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Step 1   Before you begin, check to ensure that soft reconfiguration is not configured. Your display should resemble the following:

```
wg1r1#show ip bgp neighbor 192.168.20.20 received-routes
% Inbound soft reconfiguration not enabled

wg1r1#show ip bgp neighbor 192.168.20.22 received-routes
% Inbound soft reconfiguration not enabled
```

Step 2   Enable inbound soft reconfiguration on WG*x*R1 to both neighbors "Good" and "Cheap."

Note	Using the **neighbor soft-reconfiguration** command may require your router to have much more memory than normally needed.

Exercise Verification

You have completed this exercise when you attain these results:

```
wg1r1#show ip bgp neighbor 192.168.20.20 received-routes

BGP table version is 50, local router ID is 197.1.8.1
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
*  10.0.0.0          192.168.20.233           0             0 20 i
*  128.20.0.0        192.168.20.20            0             0 20 i
*> 128.20.12.0/24    192.168.20.20            0             0 20 i
*  128.22.0.0        192.168.20.20            0             0 20 i
*> 128.22.12.0/24    192.168.20.22                          0 20 22 i
*  128.26.0.0        192.168.20.20            0             0 20 42 26 i
*  128.37.0.0        192.168.20.20            0             0 20 42 37 i
*  128.42.0.0        192.168.20.20            0             0 20 42 i
*  128.51.0.0        192.168.20.20            0             0 20 42 26 51 i
*> 128.213.0.0       192.168.20.20            0             0 20 213 i
*  128.214.0.0       192.168.20.22                          0 20 22 214 i
*  192.20.11.0       192.168.20.20            0             0 20 i
*  192.20.12.0/30    192.168.20.20            0             0 20 i
*  192.22.11.0       192.168.20.20            0             0 20 i
*  192.22.12.0/30    192.168.20.22                          0 20 22 i
*  192.26.11.0       192.168.20.20            0             0 20 42 26 i
*  192.37.11.0       192.168.20.20            0             0 20 42 37 i
*  192.42.11.0       192.168.20.20            0             0 20 42 i
*  192.51.11.0       192.168.20.20            0             0 20 42 26 51 i
*  192.213.11.0      192.168.20.20            0             0 20 213 i
*  192.214.11.0      192.168.20.22                          0 20 22 214 i
*  200.20.0.0/16     192.168.20.20            0             0 20 i
*  200.22.0.0/16     192.168.20.22                          0 20 22 i

Total number of prefixes 23
```

```
wg1r1#show ip bgp neighbor 192.168.20.22 received-routes
BGP table version is 50, local router ID is 197.1.8.1
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
*  10.0.0.0         192.168.20.233                        0 22 20 i
*> 128.20.0.0       192.168.20.22          0           100 22 i
*  128.20.12.0/24   192.168.20.20                        0 22 20 i
*> 128.22.0.0       192.168.20.22          0           100 22 i
*  128.22.12.0/24   192.168.20.22          0             0 22 i
*> 128.26.0.0       192.168.20.22          0           100 22 26 i
*> 128.37.0.0       192.168.20.22          0           100 22 26 42 37 i
*> 128.42.0.0       192.168.20.22          0           100 22 26 42 i
*> 128.51.0.0       192.168.20.22          0           100 22 26 51 i
*  128.213.0.0      192.168.20.20                        0 22 20 213 i
*> 128.214.0.0      192.168.20.22          0           100 22 214 i
*> 192.20.11.0      192.168.20.22          0           100 22 i
*> 192.20.12.0/30   192.168.20.20                     100 22 20 i
*> 192.22.11.0      192.168.20.22          0           100 22 i
*> 192.22.12.0/30   192.168.20.22          0           100 22 i
*> 192.26.11.0      192.168.20.22          0           100 22 26 i
*> 192.37.11.0      192.168.20.22          0           100 22 26 42 37 i
*> 192.42.11.0      192.168.20.22          0           100 22 26 42 i
*> 192.51.11.0      192.168.20.22          0           100 22 26 51 i
*  192.213.11.0     192.168.20.20                        0 22 20 213 i
*> 192.214.11.0     192.168.20.22          0           100 22 214 i
*> 200.20.0.0/16    192.168.20.20                     100 22 20 i
*> 200.22.0.0/16    192.168.20.22          0           100 22 i

Total number of prefixes 23
```

Answer these questions:

Q1)   What command do you use to show which entries in the BGP table of the local router
        have been propagated to a specific neighbor?

Q2)   What command do you use to display which of the routes in the local BGP table have
        been received and accepted from an indicated neighbor?

Lab Exercise 4-1: Influencing BGP Route Selection with Weights

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection by using the weight attribute in a situation where you must support connections to multiple ISPs. After completing this exercise, you will be able to meet these objectives:

■ Influence the BGP route selection process by configuring per-neighbor weights

■ Influence the BGP route selection process by configuring BGP weights with route-maps

■ Monitor BGP route selection and weights

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

■ Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

■ Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

■ Two Ethernet 10BASE-T patch cables.

■ IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

■ One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

■ Two Cisco 2610 routers with BGP-capable operating system software installed

■ One Cisco 3640 router with an installed NM-8A/S

■ Two Catalyst 2924M-XL Ethernet switches

■ Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **weight** *weight*	Assigns a weight to all updates that are received from the specified neighbor
neighbor {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	Applies a route-map to incoming or outgoing routes
route-map *name* {**permit** | **deny**} **seq**	Defines or modifies an existing entry in a route-map
match as-path *list-number*	Specifies route-map matching criteria to match an AS-path access-list
set weight *weight*	Sets the weight in a route-map
show ip bgp summary	Verifies if both BGP sessions are up
show ip bgp	Enables you to inspect the contents of the BGP table
clear ip bgp	Clears the BGP session with your neighbor

Job Aids

These job aids are available to help you complete the lab exercise:

- Currently, you are using the service provider "Cheap" as your primary provider for Internet connectivity and the service provider "Good" as your backup provider. As a result of this policy, the link toward the "Good" service provider is underused, while the link toward the provider "Cheap" is overloaded.

- An analysis of the Internet structure beyond your service providers indicates that you can improve the link use if you send traffic toward AS 213 and AS 37 directly to the "Good" service provider.

- In this exercise, you will improve the simple routing policy that you created in the lab exercise "Using Multihomed BGP Networks," by specifying BGP weights with route-maps.

- You must implement the following routing policy:

  — Prefer routes announced from router "Cheap" over those announced from router "Good"

  — Prefer routes going through or originating in AS 213 or AS 37 from router "Good"

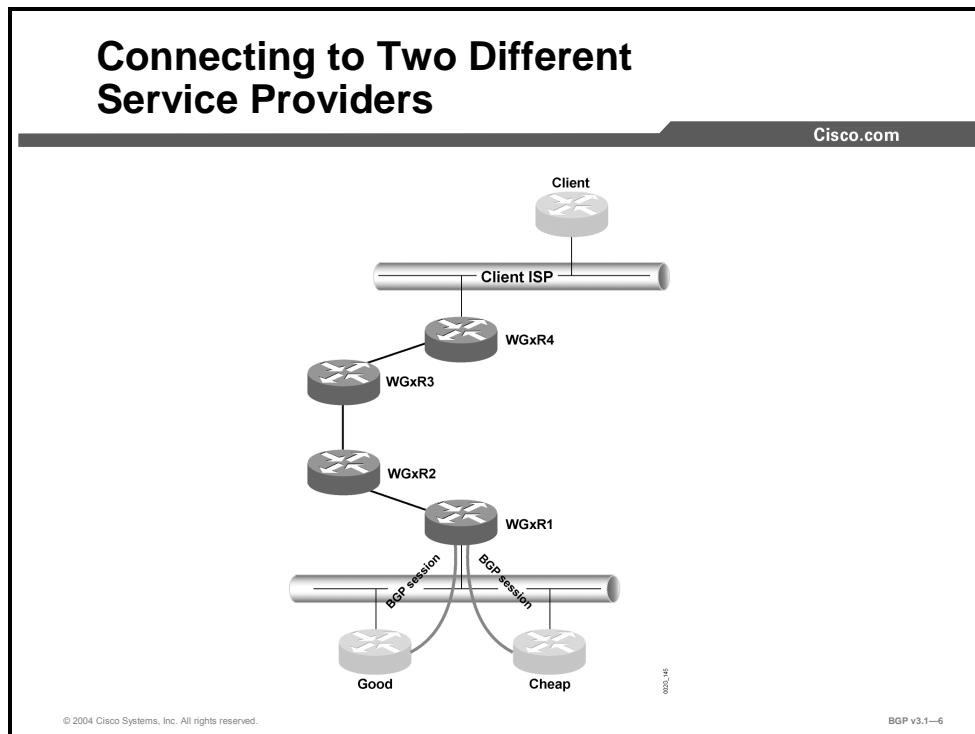- Figure 1 shows the connectivity that is established between your AS and the two service providers "Good" and "Cheap."

Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Configuring AS-path access-lists:

Step 1   Create an AS-path access-list that permits AS numbers 213 and 37 in the AS path.

Setting per-neighbor default weights:

Step 2   Set default weights for both neighboring routers. For example, use 100 for "Good" and 200 for "Cheap" as weights.

Step 3   Inspect your BGP table to verify that you prefer routes coming from "Cheap" to routes coming from "Good."

Using AS-path filters within route-maps to set weights:

Step 4   Create a new route-map. Use the previously configured AS-path access-list as the match condition in one of the route-map statements, and set the weight of matched routes as needed. Remember that routes matching this AS-path access-list from "Good" should give the highest weight; therefore, you might use 300 for weight.

Step 5   Apply the route-map to incoming updates from router "Good."

Exercise Verification

You have completed this exercise when you attain these results:

■ Verify your BGP table to see if all prefixes with two paths prefer the one through router "Cheap." Your display should resemble the following:

```
wg1r1#show ip bgp
BGP table version is 43, local router ID is 197.1.8.1
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
*  128.20.0.0       192.168.20.20          0           100 20 i
*>                  192.168.20.22          0           200 22 i
*> 128.20.12.0/24   192.168.20.20          0           100 20 i
*  128.22.0.0       192.168.20.20          0           100 20 i
*>                  192.168.20.22          0           200 22 i
*> 128.22.12.0/24   192.168.20.22                      100 20 22 i
*  128.26.0.0       192.168.20.20          0           100 20 42 26 i
*>                  192.168.20.22          0           200 22 26 i
*> 128.37.0.0       192.168.20.20          0           300 20 42 37 i
*                   192.168.20.22          0           200 22 26 42 37 i
*  128.42.0.0       192.168.20.20          0           100 20 42 i
*>                  192.168.20.22          0           200 22 26 42 i
*  128.51.0.0       192.168.20.20          0           100 20 42 26 51 i
*>                  192.168.20.22          0           200 22 26 51 i
*> 128.213.0.0      192.168.20.20          0           300 20 213 i
*> 128.214.0.0      192.168.20.22          0           200 22 214 i
*> 192.20.11.0      192.168.20.22          0           200 22 i
*> 192.20.12.0/30   192.168.20.20                      200 22 20 i
*> 192.22.11.0      192.168.20.22          0           200 22 i
*> 192.22.12.0/30   192.168.20.22          0           200 22 i
*> 192.26.11.0      192.168.20.22          0           200 22 26 i
*> 192.37.11.0      192.168.20.22          0           200 22 26 42 37 i
*> 192.42.11.0      192.168.20.22          0           200 22 26 42 i
*> 192.51.11.0      192.168.20.22          0           200 22 26 51 i
*> 192.168.1.0      0.0.0.0                0         32768 i
*> 192.214.11.0     192.168.20.22          0           200 22 214 i
*> 197.1.0.0/16     0.0.0.0                          32768 i
*  200.20.0.0/16    192.168.20.20          0           100 20 i
*>                  192.168.20.20                      200 22 20 i
*  200.22.0.0/16    192.168.20.22                      100 20 22 i
*>                  192.168.20.22          0           200 22 i
```

Answer these questions:

Q1)   Did all paths automatically get a weight of 100 or 200? Why not? What did you have to do?

Q2)   Name some parameters and attributes that are used for best-path selection.

Lab Exercise 4-2: Setting BGP Local Preference

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection using the local preference attribute in a situation where you must support multiple connections to an ISP. After completing this exercise, you will be able to meet these objectives:

- Configure BGP local preference using route-maps

- Monitor BGP local preference

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
no synchronization	Disables BGP synchronization
neighbor {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	Applies a route-map to incoming or outgoing routing updates
set local-preference *number*	Sets the local preference attribute within a route-map
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
clear ip bgp	Restarts the BGP session with your BGP neighbor

Job Aids

These job aids are available to help you complete the lab exercise:

- You want to have two links with the "Good" service provider to increase the reliability of your Internet service. With several links that connect you to the same service provider, you must use local preference in your AS to ensure consistent AS-wide routing policy.

- In this exercise, you will establish a second link toward the "Good" service provider and use the local preference attribute to select the newly established link as the preferred exit point from your network.

- The additional link that you establish will connect WG*x*R2 and the "Good" router through the Frame Relay network. You will configure an EBGP session between WG*x*R2 and "Good" over this link. All traffic from your AS toward "Good" should flow over this link.

- On WG*x*R2, use data link connection identifier (DLCI) 20*x* and IP address 192.168.3*x*.1/30 for the Frame Relay link connection. The other side of this permanent virtual circuit (PVC) is connected to router "Good," which is already configured.

- The BGP routing design contains the following items:

   — AS *x* should prefer AS 20 as the upstream service provider. Router WG*x*R2 should be used as the exit point under normal circumstances.

   — Peering to AS 20 through WG*x*R1 should be used only if the primary link fails.

- Figure 1 shows the new physical connectivity, BGP sessions, and the expected traffic flow in the network.
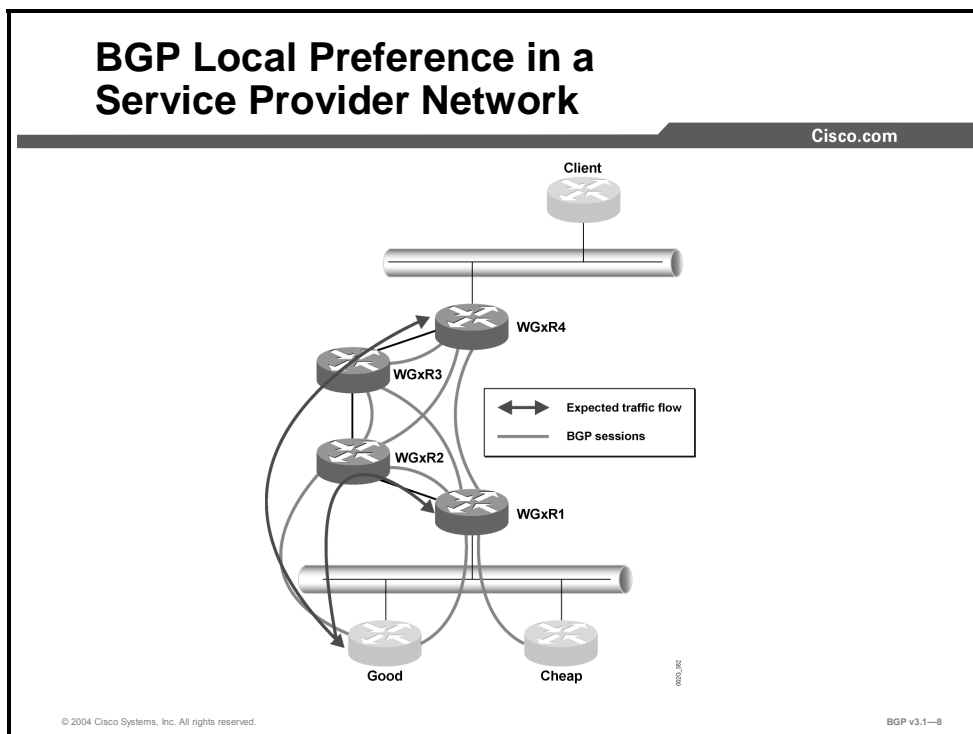

Figure 1: BGP Local Preference in a Service Provider Network

Exercise Procedure

Complete these steps:

Configuring additional IP connectivity:

Step 1   Create another point-to-point subinterface on WG*x*R2 using the parameters from the following table.

Router	IP Address	DLCI
WG*x*R2	192.168.3*x*.1/30	20*x*

Establishing EBGP peering between WG*x*R2 and router "Good":

Step 2   Configure the router "Good" as the EBGP neighbor by using the parameters from the following table.

Parameter	Value
Service provider IP address	192.168.3*x*.2
Service provider AS number	20

Establishing full-mesh IBGP:

Step 3   Establish IBGP sessions between all routers in your pod with disabled synchronization. Remember to use the good practice of implementing a stable loopback interface address for peering.

Step 4   Remember that external networks are not carried in your Interior Gateway Protocol (IGP). Therefore, you will need the **next-hop-self** command.

Using local preference to ensure AS-wide routing policy:

Step 5   On WG*x*R1, remove neighbor weight statements and any route-map statements that modify the weight attribute on WG*x*R1 for the routers "Good" and "Cheap."

Step 6   Create a new route-map on router WG*x*R2. Within the route map, set the local preference higher than the default value of 100 (for example, 200).

Step 7   Apply the route-map to incoming updates from router "Good."

Step 8   Perform clearing of the BGP session between the routers by using the **clear ip bgp *** command. Use a soft clearing of WG*x*R1 if you still have soft inbound reconfiguration enabled.

Exercise Verification

You have completed this exercise when you attain these results:

- On WG*x*R2, verify that the local preference has been changed and also note the IBGP routes. Your display should resemble the following:

```
wg1r2#show ip bgp
BGP table version is 29, local router ID is 197.1.3.1
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
*> 10.0.0.0         192.168.31.2            0    200      0 20 i
*> 128.20.0.0       192.168.31.2            0    200      0 20 i
*> 128.20.12.0/24   192.168.31.2            0    200      0 20 i
*> 128.22.0.0       192.168.31.2            0    200      0 20 i
*> 128.22.12.0/24   192.168.31.2                 200      0 20 22 i
*> 128.26.0.0       192.168.31.2            0    200      0 20 42 26 i
*> 128.37.0.0       192.168.31.2            0    200      0 20 42 37 i
*> 128.42.0.0       192.168.31.2            0    200      0 20 42 i
*> 128.51.0.0       192.168.31.2            0    200      0 20 42 26 51 i
*> 128.213.0.0      192.168.31.2            0    200      0 20 213 i
*> 128.214.0.0      192.168.31.2                 200      0 20 22 214 i
*> 192.20.11.0      192.168.31.2            0    200      0 20 i
*> 192.20.12.0/30   192.168.31.2            0    200      0 20 i
*> 192.22.11.0      192.168.31.2            0    200      0 20 i
*> 192.22.12.0/30   192.168.31.2                 200      0 20 22 i
*> 192.26.11.0      192.168.31.2            0    200      0 20 42 26 i
*> 192.37.11.0      192.168.31.2            0    200      0 20 42 37 i
*> 192.42.11.0      192.168.31.2            0    200      0 20 42 i
*> 192.51.11.0      192.168.31.2            0    200      0 20 42 26 51 i
*  i192.168.1.0     192.168.1.1             0    100      0 i
*>                  0.0.0.0                 0          32768 i
*>i192.168.31.0     192.168.1.1             0    100      0 i
*> 192.213.11.0     192.168.31.2            0    200      0 20 213 i
*> 192.214.11.0     192.168.31.2                 200      0 20 22 214 i
*>i197.1.0.0/21     192.168.1.1                  100      0 i
*>i197.1.0.0/16     192.168.1.1                  100      0 i
*>i197.1.8.0/22     192.168.1.1                  100      0 i
*> 200.20.0.0/16    192.168.31.2            0    200      0 20 i
*> 200.22.0.0/16    192.168.31.2                 200      0 20 22 I
```

- Use the **traceroute** command from router WG*x*R4 and WG*x*R1 to 192.20.11.1. Your display should resemble the following:

```
WG1R4#traceroute 192.20.11.1
Type escape sequence to abort.
Tracing the route to 192.20.11.1

  1 192.168.1.9 32 msec 24 msec 20 msec
  2 192.168.1.5 32 msec 45 msec 40 msec
  3 192.168.31.2 56 msec *  52 msec

wg1r1#traceroute 192.20.11.1
Type escape sequence to abort.
Tracing the route to 192.20.11.1

  1 192.168.1.2 40 msec 28 msec 24 msec
  2 192.168.31.2 12 msec *  16 msec
```

- Use **traceroute** from router "Good" to interface loopback0 on routers WG*x*R4 and WG*x*R1. (Note that this result might depend on the age of the EBGP sessions from step 8 above [oldest EBGP path that was selected] and necessitate clearing of the router "Good" peer with WG*x*R2 to demonstrate asymmetrical routing. The next lab will address these issues.) Your display should resemble the following:

```
Good#traceroute 197.1.7.1
Type escape sequence to abort.
Tracing the route to 197.1.7.1

  1 wg1 (192.168.20.1) 4 msec 0 msec 4 msec
  2 192.168.1.2 [AS 1] 20 msec 28 msec 24 msec
  3 192.168.1.6 [AS 1] 40 msec 45 msec 36 msec
  4 192.168.1.10 [AS 1] 52 msec *  48 msec

Good#traceroute 197.1.8.1
Type escape sequence to abort.
Tracing the route to 197.1.8.1

  1 wg1 (192.168.20.1) 4 msec *  0 msec
```

- Compare the two outputs of **traceroute** to determine if the routing is symmetrical. Both **traceroute** commands should show router WG*x*R2 in the path.

Answer these questions:

Q1)   Is routing between router WG*x*R1 and "Good" symmetrical?

Q2)   Which routers receive the local preference attribute?

Lab Exercise 4-3: Understanding BGP Multi-Exit Discriminators

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection using the MED attribute in a situation where you must support multiple connections to an ISP. After completing this exercise, you will be able to meet these objectives:

- Configure the BGP MED using route-maps

- Monitor the BGP MED

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor {*ip-address* | *peer-group-name*} **route-map** *map-name* {**in** | **out**}	Applies a route-map to incoming or outgoing routing updates
route-map *name* {**permit** | **deny**} **seq**	Defines or modifies an existing entry in a route-map
set metric *metric*	Sets the MED in a route-map
clear ip bgp *	Resets the BGP peer
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command

Job Aids

These job aids are available to help you complete the lab exercise:

- You have noticed that the traffic from router "Good" toward your AS sometimes passes through the router WG*x*R1, resulting in asymmetrical routing. You will use the MED to indicate to the "Good" router which exit point that it should use.

Note   This exercise is a continuation of the previous lab exercise.

- The implementation should ensure symmetrical routing. You should use backup peering only when the primary link is down. Use the MED to influence neighboring AS 20 to choose the preferred return path.

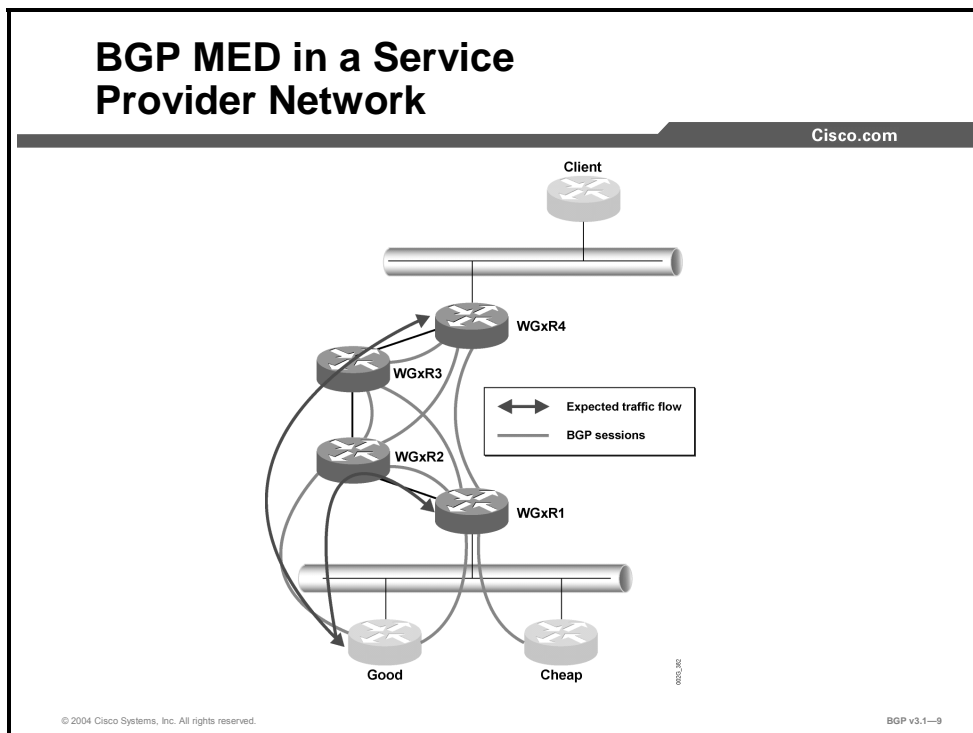- Figure 1 shows the physical connectivity, BGP sessions, and traffic flow in the network.

Figure 1: BGP MED in a Service Provider Network

Exercise Procedure

Complete these steps:

Step 1   Before starting the exercise, check the BGP table for router "Good" to see if the MED is present in BGP updates that are received from WG*x*R1 and WG*x*R2.

Configuring WG*x*R1 to set the MED:

Step 2   Create a new route-map, which sets the BGP MED to a high value (for example, 200). Apply the route-map to outgoing updates toward the router "Good."

Configuring WG*x*R2 to set the MED:

Step 3   Create a new route-map, which sets the BGP MED to a low value (for example, 100). Apply the route-map to outgoing updates toward the router "Good."

Exercise Verification

You have completed this exercise when you attain these results:

- Check the BGP table on router "Good" to see if the MED is present in BGP updates that are received from WG*x*R1 and WG*x*R2. Your display should resemble the following:

```
Good#show ip bgp
BGP table version is 46, local router ID is 199.199.199.199
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
*> 10.0.0.0         192.168.20.233           0          32768 i
*  128.20.0.0       192.168.20.22            0              0 22 i
*>                  0.0.0.0                  0          32768 i
*> 128.20.12.0/24   0.0.0.0                  0          32768 i
*  128.22.0.0       192.168.20.22            0              0 22 i
*>                  0.0.0.0                  0          32768 i
*> 128.22.12.0/24   192.168.20.22            0              0 22 i
*  128.26.0.0       192.168.20.22            0              0 22 26 i
*>                  0.0.0.0                  0          32768 i
*  128.37.0.0       192.168.20.22            0              0 22 26 42 37 i
*>                  0.0.0.0                  0          32768 i
*  128.42.0.0       192.168.20.22            0              0 22 26 42 i
*>                  0.0.0.0                  0          32768 i
*  128.51.0.0       192.168.20.22            0              0 22 26 51 i
*>                  0.0.0.0                  0          32768 i
*> 128.213.0.0      0.0.0.0                  0          32768 i
*> 128.214.0.0      192.168.20.22            0              0 22 214 i
*  192.20.11.0      192.168.20.22            0              0 22 i
*>                  0.0.0.0                  0          32768 i
*> 192.20.12.0/30   0.0.0.0                  0          32768 i
*  192.22.11.0      192.168.20.22            0              0 22 i
*>                  0.0.0.0                  0          32768 i
*> 192.22.12.0/30   192.168.20.22            0              0 22 i
*  192.26.11.0      192.168.20.22            0              0 22 26 i
*>                  0.0.0.0                  0          32768 i
*  192.37.11.0      192.168.20.22            0              0 22 26 42 37 i
*>                  0.0.0.0                  0          32768 i
*  192.42.11.0      192.168.20.22            0              0 22 26 42 i
*>                  0.0.0.0                  0          32768 i
*  192.51.11.0      192.168.20.22            0              0 22 26 51 i
*>                  0.0.0.0                  0          32768 i
*  192.168.1.0      192.168.20.1                            0 22 1 i
*                   192.168.20.1           200              0 1 i
```

```
*>                     192.168.31.1         100              0 1 i
*   192.168.31.0       192.168.20.1                          0 22 1 i
*>                     192.168.31.1         100              0 1 i
*                      192.168.20.1         200              0 1 i
*> 192.213.11.0        0.0.0.0                0             32768 i
*> 192.214.11.0        192.168.20.22          0               0 22 214 i
*   197.1.0.0/21       192.168.20.1                          0 22 1 i
*>                     192.168.31.1         100              0 1 i
*                      192.168.20.1         200              0 1 i
*   197.1.0.0/16       192.168.20.1                          0 22 1 i
*>                     192.168.31.1         100              0 1 i
*                      192.168.20.1         200              0 1 i
*   197.1.8.0/22       192.168.20.1                          0 22 1 i
*>                     192.168.31.1         100              0 1 i
*                      192.168.20.1         200              0 1 i
*> 200.20.0.0/16       0.0.0.0                0             32768 i
*> 200.22.0.0/16       192.168.20.22          0               0 22 I
```

■ Use the **traceroute** command from routers WG*x*R4 and WG*x*R1 to 192.20.11.1. Your display should resemble the following:

```
wg1r4#traceroute 192.20.11.1
Type escape sequence to abort.
Tracing the route to 192.20.11.1

  1 192.168.1.9 32 msec 24 msec 20 msec
  2 192.168.1.5 28 msec 44 msec 41 msec
  3 192.168.31.2 60 msec 64 msec 56 msec


wg1r1#traceroute 192.20.11.1
Type escape sequence to abort.
Tracing the route to 192.20.11.1

  1 192.168.1.2 40 msec 28 msec 20 msec
  2 192.168.31.2 32 msec *  36 msec
```

■ Use the **traceroute** command from router "Good" to interface loopback1 on routers WG*x*R4 and WG*x*R1. Your display should resemble the following:

```
Good#traceroute 197.1.7.1
Type escape sequence to abort.
Tracing the route to 197.1.7.1

  1 192.168.31.1 [AS 1] 36 msec 28 msec 24 msec
  2 192.168.1.6 [AS 1] 32 msec 36 msec 36 msec
  3 192.168.1.10 [AS 1] 52 msec *  48 msec

Good#traceroute 197.1.8.1
Type escape sequence to abort.
Tracing the route to 197.1.8.1


  1 192.168.31.1 [AS 1] 36 msec 28 msec 24 msec
  2 192.168.1.1 [AS 1] 40 msec *  36 msec
```

■ Compare the two outputs of **traceroute** to determine if the routing is symmetrical. Both **traceroute** commands should show router WG*x*R2 in the path.

Answer these questions:

Q1)   Which parameters and attributes have to be equal before the MED is compared to select the best path?

Q2)   What is the default value of the MED?

Lab Exercise 4-4: Addressing BGP Communities

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to influence route selection by using the BGP community attribute in a situation where you must support multiple connections to an ISP. After completing this exercise, you will be able to meet these objectives:

- Configure route tagging by using BGP communities

- Configure BGP community propagation

- Monitor BGP communities

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor {*ip-address*	*peer-group-name*} **route-map** *map-name* {**in**	**out**}	Applies a route-map to incoming or outgoing routing updates
route-map *name* {**permit**	**deny**} **seq**	Defines or modifies an existing entry in a route-map
match community *c-acl*	Matches communities in a route-map by using a community-list
set community *community* [**additive**]	Sets community attributes in a route-map—use the keyword **additive** to append communities instead of replacing them
set local-preference *num*	Sets local preference attribute
ip bgp-community new-format	Configures BGP communities using as:nn format
ip community-list *num* {**permit**	**deny**} **community**	Creates a community access-list
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
show ip bgp community [**community** [**community** …]] [**exact-match**]	Enables you to view BGP routes that have at least one community attribute or those that are specified in the command
show ip bgp community-list *c-list* [**exact-match**]	Enables you to view BGP routes that are permitted by the specified community-list
neighbor *ip-address* **send-community**	Enables community propagation to BGP neighbors
neighbor *ip-address* **shutdown**	Disables communication with a BGP neighbor

Job Aids

These job aids are available to help you complete the lab exercise:

■ Your second service provider "Cheap" has upgraded its link to your network to match that of the provider "Good." You decide to change your route selection policy to use the newly updated link from "Cheap." As a result of the new service that is available from "Cheap," you decide to remove the backup link to provider "Good" because you can now achieve both link and provider redundancy.

■ You wish for all outgoing traffic to still use the link to "Good," but all return traffic should use the newly updated link from "Cheap." Knowing that you have a connection to the provider "Good," you still wish to use the primary connection to "Good" for return traffic should the link to provider "Cheap" fail.

■ You cannot influence return path selection across multiple providers and solicit assistance from provider "Good." Provider "Good" agrees to set local preference for any routes that you mark as follows:

— Local preference of 50 for routes marked with a community attribute of x:20

■ Figure 1 shows the physical connectivity, BGP sessions, and traffic flow in the network.
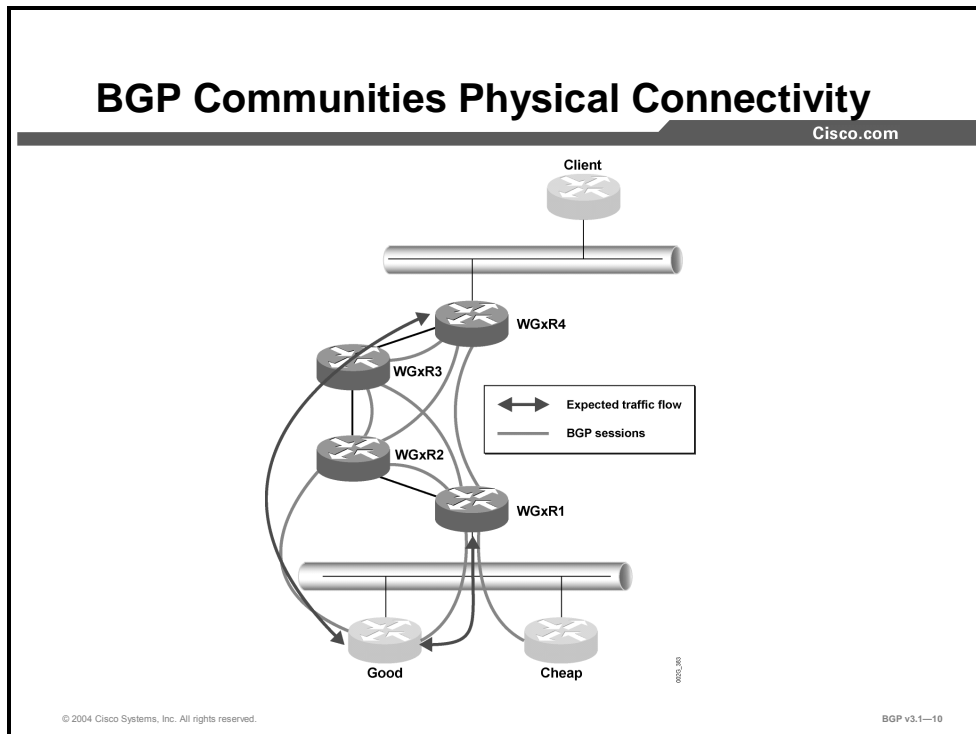
Figure 1: BGP Communities Physical Connectivity

Exercise Procedure

Complete these steps:

Step 1   Before starting this lab, use the **traceroute** command from router "Good" to
197.*x*.7.1. The trace should match the display here (where *x* is your workgroup
number):

```
Good#traceroute 197.x.7.1

Type escape sequence to abort.
Tracing the route to 197.x.7.1

1 192.168.3x.1 [AS 1] 16 msec 16 msec 16 msec
2 192.168.x.6 [AS 1] 28 msec 32 msec 28 msec
3 192.168.x.10 [AS 1] 44 msec *  40 msec
```

Step 2   Remove the BGP neighbor statement on WG*x*R1 for router "Good" or simply
disable communication via the **neighbor shutdown** command.

Configuring BGP community propagation:

Step 3   Configure BGP community propagation on router WG*x*R2.

Step 4   Enable your WG*x*R2 router to configure communities using the as: nn format.

Configuring outbound filters on WG*x*R2:

Step 5   Create a route-map to set the community attribute on all BGP updates that are sent
from WG*x*R2 to the provider "Good." Set the community attribute to *x*:20, where *x*
is your workgroup number.

Step 6   Apply the route-map to the BGP neighbor "Good" in the outbound direction.

Exercise Verification

You have completed this exercise when you attain these results:

■ Log onto the router "Good" and verify that the BGP community attribute has been correctly
set. Your display should resemble the following:

```
Good>show ip bgp community 1:20

Good#show ip bgp community 1:20
BGP table version is 61, local router ID is 199.199.199.199
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
*  192.168.1.0      192.168.31.1             0     50      0 1 i
*  197.1.0.0/16     192.168.31.1                   50      0 1 i
```

■ Log onto router "Good" and perform the same traceroute that was done before you started
the configuration in this lab. Your display should resemble the following:

```
Good#traceroute 197.1.7.1
Type escape sequence to abort.
Tracing the route to 197.1.7.1

  1 wg1 (192.168.20.1) 0 msec 0 msec 0 msec
  2 192.168.1.2 [AS 1] 16 msec 16 msec 16 msec
  3 192.168.1.6 [AS 1] 28 msec 28 msec 36 msec
  4 192.168.1.10 [AS 1] 44 msec *  44 msec
```

- On router "Good," execute the command: **show ip bgp 197.1.7.1**. Your display should resemble the following:

```
Good#show ip bgp 197.1.7.1
BGP routing table entry for 197.1.0.0/21, version 22
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to peer-groups:
    students2
  1, (aggregated by 1 197.1.8.1)
    192.168.31.1 from 192.168.31.1 (197.1.3.1)
      Origin IGP, localpref 50, valid, external, atomic-aggregate
      Community: 1:20

  22 1, (aggregated by 1 197.1.8.1)
    192.168.20.1 from 192.168.20.22 (192.20.11.1)
      Origin IGP, localpref 100, valid, external, atomic-aggregate, best
```

Note that the best path, because of local preference, is the bottom route, and the next hop is 192.168.20.22 ("Cheap"), but because of the shared media, the next hop has not been changed. This result is true even though the AS-path length is longer, because local preference is a stronger route selection tool.

Answer these questions:

Q1)   What do you have to do to enable community propagation?

Q2)   What mechanisms can you use to match or set communities?

Lab Exercise 6-1: Introducing Route Reflectors

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will enable route reflectors to modify IBGP split-horizon rules in a transit AS. After completing this exercise, you will be able to meet these objectives:

- Plan the migration of an existing IBGP backbone to a backbone with route reflectors

- Configure BGP route reflectors on a BGP backbone

- Monitor a BGP backbone that contains route reflectors

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
bgp cluster-id *id*	Specifies the cluster-ID
neighbor *ip-address* **route-reflector-client**	Establish an IBGP session by using your workgroup number as the AS number
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command

Job Aids

These job aids are available to help you complete the lab exercise:

- In this exercise, you will replace the IBGP full mesh with a hierarchy of BGP route reflectors.

- Configure IBGP sessions in your AS according to this design:

  — Cluster #1 is your top-level cluster. Within this cluster, router WG*x*R3 is the route reflector, and WG*x*R2 is the client.

  — WG*x*R2 is the route reflector in cluster #2, and WG*x*R1 is its client.

  — Router WG*x*R4 is not participating in any cluster and needs to be in IBGP full mesh with top-level route reflectors.

- The required IBGP sessions and the cluster structure are outlined in Figure 1.
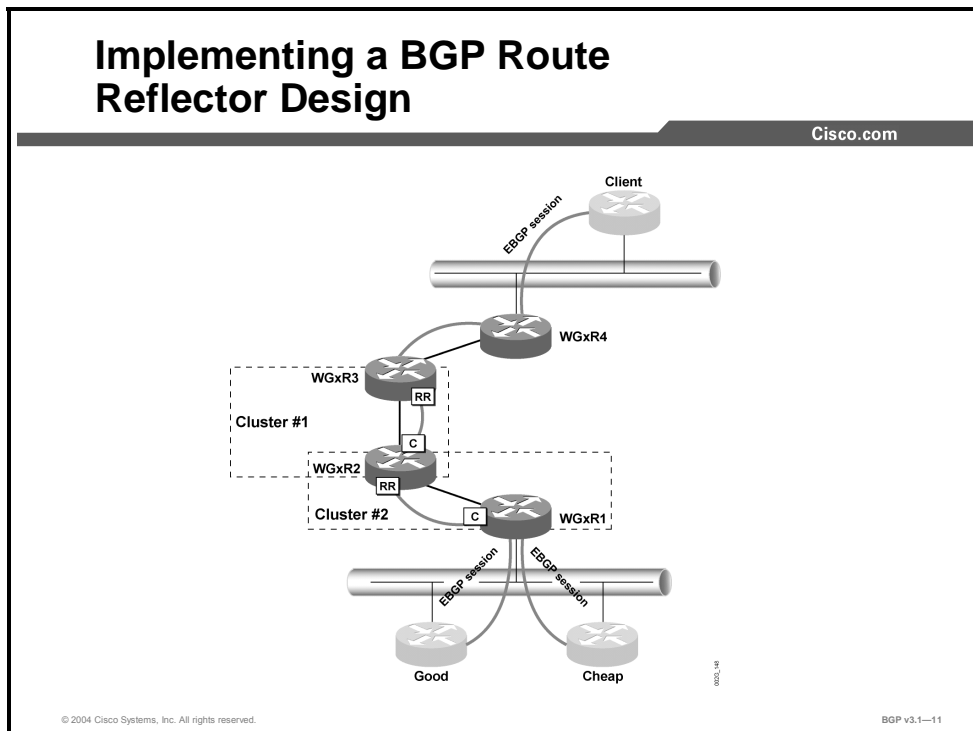

Figure 1: Implementing a BGP Route Reflector Design

Exercise Procedure

Complete these steps:

Cleaning up your BGP configurations:

Step 1 Remove the EBGP session on WG*x*R2 to router "Good." Remember to remove next-hop-self because EBGP is no longer configured.

Step 2 Enable the shutdown neighbor from WG*x*R1 to "Good" or reconfigure if the neighbor statement was removed.

Step 3 Configure EBGP from WG*x*R4 to router "Client" as 192.168.21.99 with AS 99. Remember that the external network is not announced in your IGP.
Option: WG*x*R1 filters to "Good" and "Cheap" can be removed or left intact.

Configuring the top-level route reflector WG*x*R3:

Step 4 Configure cluster-ID 101 on the top-level route reflector WG*x*R3. Configure the client on WG*x*R3 and remove unnecessary IBGP sessions.

Configuring the second-level route reflector WG*x*R2:

Step 5 Configure cluster-ID 102 on WG*x*R2. Configure the client on WG*x*R2 and remove unnecessary IBGP sessions from WG*x*R2.

Removing unnecessary BGP neighbors from your AS:

Step 6 Remove the IBGP sessions that are no longer needed from routers in your pod.

Exercise Verification

You have completed this exercise when you attain these results:

- Check your BGP connections on the core routers and ensure that they are connected only to the peers, as shown in the lab diagram. Your display should resemble the following:

```
WG1R1#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.20.20   4    20    5923    4224      138    0    0 01:11:00        23
192.168.20.22   4    22    1720    1705      138    0    0 01:11:03        23
197.1.2.1       4     1    1665    1703      138    0    0 00:02:52        11


WG1R2#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
197.1.1.1       4     1    4149    4053     1074    0    0 00:03:19        25
197.1.4.1       4     1    4035    4053     1074    0    0 00:01:05        13


WG1R3#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
197.1.2.1       4     1    4053    4035     1028    0    0 00:01:26        25
197.1.6.1       4     1    4171    4068     1028    0    0 01:27:02        13


WG1R4#show ip bgp summary
…
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.21.99   4    99    1744    1713      225    0    0 00:46:07        11
197.1.4.1       4     1    1674    1745      225    0    0 01:27:19        23
```

- Check to see if router "Client" is receiving networks from your network. The command output will vary depending on filters. Your display should resemble the following:

```
Client#show ip bgp
BGP table version is 210, local router ID is 197.99.111.1
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
*> 99.0.0.0         0.0.0.0                 0         32768 i
*> 128.20.0.0       192.168.21.1                          0 1 22 i
*> 128.20.12.0/24   192.168.21.1                          0 1 20 i
*> 128.22.0.0       192.168.21.1                          0 1 22 i
*> 128.22.12.0/24   192.168.21.1                          0 1 20 22 i
*> 128.26.0.0       192.168.21.1                          0 1 22 26 i
*> 128.37.0.0       192.168.21.1                          0 1 20 42 37 i
*> 128.42.0.0       192.168.21.1                          0 1 20 42 i
*> 128.51.0.0       192.168.21.1                          0 1 22 26 51 i
*> 128.213.0.0      192.168.21.1                          0 1 20 213 i
*> 128.214.0.0      192.168.21.1                          0 1 22 214 i
*> 192.20.11.0      192.168.21.1                          0 1 22 i
*> 192.20.12.0/30   192.168.21.1                          0 1 22 20 i
*> 192.22.11.0      192.168.21.1                          0 1 22 i
*> 192.22.12.0/30   192.168.21.1                          0 1 22 i
*> 192.26.11.0      192.168.21.1                          0 1 22 26 i
*> 192.37.11.0      192.168.21.1                          0 1 22 26 42 37 i
*> 192.42.11.0      192.168.21.1                          0 1 22 26 42 i
*> 192.51.11.0      192.168.21.1                          0 1 22 26 51 i
*> 192.168.1.0      192.168.21.1                          0 1 i
*> 192.214.11.0     192.168.21.1                          0 1 22 214 i
*> 197.1.0.0/16     192.168.21.1                          0 1 i
*> 197.99.1.0       0.0.0.0                 0         32768 i
*> 197.99.11.0      0.0.0.0                 0         32768 i
*> 197.99.12.0      0.0.0.0                 0         32768 i
*> 197.99.13.0      0.0.0.0                 0         32768 i
*> 197.99.20.0      0.0.0.0                 0         32768 i
*> 197.99.22.0      0.0.0.0                 0         32768 i
*> 197.99.80.0      0.0.0.0                 0         32768 i
*> 197.99.111.1/32  0.0.0.0                 0         32768 i
*> 197.99.120.0     0.0.0.0                 0         32768 i
*> 197.99.128.0/20  0.0.0.0                 0         32768 i
*> 200.20.0.0/16    192.168.21.1                          0 1 20 i
*> 200.22.0.0/16    192.168.21.1                          0 1 22 i
```

- Check to see if the cluster-list attribute is applied to routes passing through the AS. Your display should resemble the following:

```
WG1R1#show ip bgp 197.99.1.0
BGP routing table entry for 197.99.1.0/24, version 129
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  192.168.20.20 192.168.20.22
  99
    192.168.21.99 (metric 202) from 197.1.2.1 (197.1.7.1)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 197.1.7.1, Cluster list: 0.0.0.102, 0.0.0.101
```

Answer these questions:

Q1) Did this design require you to configure a cluster-ID?

Q2) What is the default cluster-ID?

Q3) When do you have to configure a cluster-ID?

Lab Exercise 6-2: Configuring and Monitoring Confederations

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will enable confederations to modify IBGP AS-path processing in a transit AS. After completing this exercise, you will be able to meet these objectives:

- Plan the migration of an existing IBGP backbone toward BGP confederations

- Configure BGP confederations on a BGP backbone

- Monitor a BGP backbone containing BGP confederations

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10-BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
bgp confederation identifier *x*	Specifies the real AS number
bgp confederation peers *as1* [*as2* [*as3*] ...]	Lists the neighboring intraconfederation autonomous systems
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
neighbor *ip-address* **remote-as** *as-number*	Configures a BGP neighbor
neighbor *ip-address* **update-source** *interface*	Configures the source interface of a BGP session
neighbor *ip-address* **ebgp-multihop**	Configures an EBGP session between neighbors that are not directly connected

Job Aids

These job aids are available to help you complete the lab exercise:

■ The AS numbers that will be used in the confederation, as well as the BGP sessions that must be established, are shown in Figure 1.

■ To successfully complete this exercise, you must remove all BGP processes and start new processes with private AS numbers according to the following table.

Router	Member-AS Number
WG*x*R1	AS 65001
WG*x*R2, WG*x*R3	AS 65002
WG*x*R4	AS 65003

■ Routers in member autonomous systems should originate their own networks. The sessions should be established on the routers between Loopback0 interfaces as shown in Figure 1.

■ AS 99 should receive only prefixes that originate in your AS *x* and AS 37. You should accept only prefixes that originate in AS 99 from router "Client."

Implementing a BGP Confederation Design

Cisco.com

© 2004 Cisco Systems, Inc. All rights reserved.

BGP v3.1—12
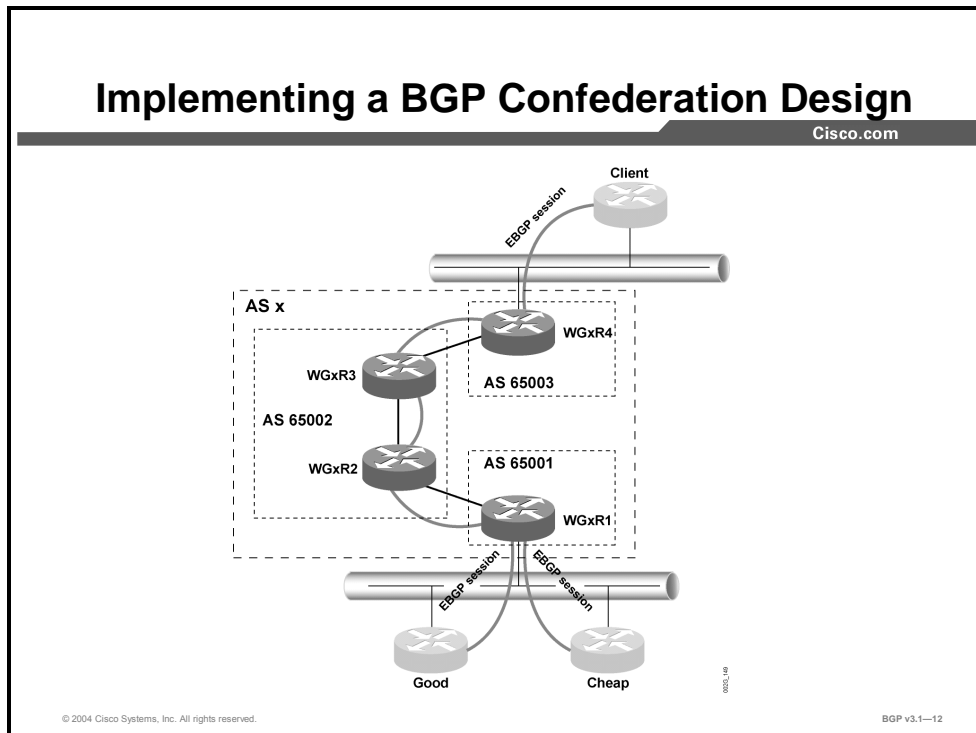
Figure 1: Implementing a BGP Confederation Design

Exercise Procedure

Complete these steps:

Perform the following steps on all routers in your workgroup:

Step 1   Remove the BGP process and start a new BGP process with the proper member-AS number.

Step 2   Announce networks that originate in this subautonomous system. Also announce network 192.168.*x*.0/24.

Step 3   Set the confederation identifier by using your real AS number. List all intraconfederation neighboring autonomous systems by using the **bgp confederation peers** command.

Step 4   Establish intraconfederation EBGP sessions and IBGP sessions within the member-AS. Use loopback interfaces for all BGP sessions.

Perform the following steps only on the specific router that is indicated in each step:

Step 5   On WG*x*R1, configure routers "Good" and "Cheap" as your BGP neighbors.

Step 6   On WG*x*R4, configure router "Client" as your BGP neighbor.

Step 7   On WG*x*R4, apply AS-path filters to incoming and outgoing updates toward router "Client."

Note	The filter that permits networks that originate in your AS needs to be changed to permit all prefixes that originate in all intraconfederation autonomous systems.

Exercise Verification

You have completed this exercise when you attain these results:

- Check to see if router "Client" is receiving only your networks and those originating in AS 37. Your display should resemble the following:

```
Client#show ip bgp regexp ^1_
BGP table version is 319, local router ID is 197.99.111.1
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
*> 128.37.0.0       192.168.21.1                       0 1 20 42 37 i
*> 192.37.11.0      192.168.21.1                       0 1 20 42 37 i
*> 192.168.1.0      192.168.21.1          0            0 1 i
*> 197.1.1.0        192.168.21.1                       0 1 i
*> 197.1.2.0        192.168.21.1                       0 1 i
*> 197.1.3.0        192.168.21.1                       0 1 i
*> 197.1.4.0        192.168.21.1                       0 1 i
*> 197.1.5.0        192.168.21.1                       0 1 i
*> 197.1.6.0        192.168.21.1          0            0 1 i
*> 197.1.7.0        192.168.21.1          0            0 1 i
*> 197.1.8.0        192.168.21.1                       0 1 i
```

- Use the **show ip route ospf** command to make sure that your IGP carries only your internal networks.

Answer these questions:

Q1)   What additional command did you have to use to establish intraconfederation EBGP
        sessions between loopback interfaces?

Q2)   What is necessary in the AS-path filters?

Lab Exercise 7-1: Limiting the Number of Prefixes Received from a BGP Neighbor

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will configure BGP to limit the number of prefixes that are received from a neighbor. After completing this exercise, you will be able to meet these objectives:

- Configure the BGP maximum-prefix function
- Monitor the BGP maximum-prefix function

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.
- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.
- Two Ethernet 10BASE-T patch cables.
- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed
- Two Cisco 2610 routers with BGP-capable operating system software installed
- One Cisco 3640 router with an installed NM-8A/S
- Two Catalyst 2924M-XL Ethernet switches
- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *ip-address* **maximum-prefix** *maximum* [**threshold**] [**warning-only**][**restart** *restart-interval*]	Specifies the allowed maximum number of prefixes
show ip bgp	Enables you to inspect the contents of the BGP table
show ip neighbor *neighbor*	Shows detailed information about a specific neighbor
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command

Job Aids

These job aids are available to help you complete the lab exercise:

■ You have noticed that one of your upstream service providers occasionally announces a very large number of BGP prefixes, which results in memory and CPU overload on your router (WGxR1). You suspect that the service provider might have internal problems, and you would like to be protected against these problems. The service provider is asking for evidence that you are receiving more prefixes than you should.

■ In this exercise, you will configure logging on your router to warn you if you receive more BGP prefixes than you should from your service provider.

■ Your analysis shows that you must perform the following tasks to minimize the size of the BGP table on router WGxR1:

— Create a warning log whenever your service provider ("Cheap") sends you more than 15 prefixes.

■ Figure 1 shows the connectivity that is established between your AS and the two service providers "Good" and "Cheap."
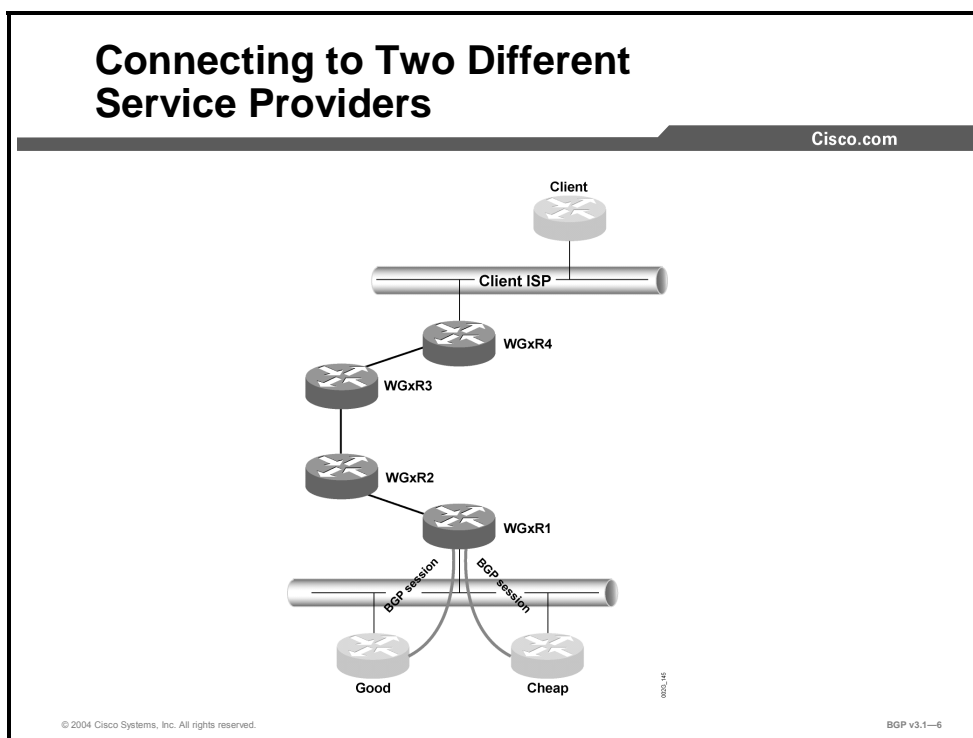
Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Configuring the maximum-prefix function:

Step 1   Set a limit to the number of prefixes that your router WG*x*R1 accepts from router "Cheap." Do not disable BGP, but instead create a warning message in the router log.

Step 2   Check your console for the following two messages:

```
22:46:40: %BGP-4-MAXPFX: No. of prefix received from 192.168.20.22 (afi 0) reaches 12,
max 15
22:49:00: %BGP-3-MAXPFXEXCEED: No. of prefix received from 192.168.20.22 (afi 0): 16
exceed limit 15
```

Step 3   After showing the service provider your logs, you have decided to take administrative action by implementing a BGP policy that will terminate the neighbor session between you and your upstream provider ("Cheap") should it continue to violate your agreement on the number of prefixes that are advertised to your network. Configure your router such that the BGP session is terminated if your router (WG*x*R1) receives more than the agreed number of prefixes.

Exercise Verification

You have completed this exercise when you attain these results:

- Verify if both neighbors are up. If not, find out why. Your display should resemble the following:

```
BGP-4-MAXPFX: No. of prefix received from 192.168.20.22 (afi 0) reaches 12, max 15
BGP-3-MAXPFXEXCEED: No. of prefix received from 192.168.20.22 (afi 0): 16 exceed limit
15
BGP-5-ADJCHANGE: neighbor 192.168.20.22 Down BGP Notification sent
BGP-3-NOTIFICATION: sent to neighbor 192.168.20.22 3/1 (update malformed) 0 bytes

WG1R1#show ip bgp summary
BGP router identifier 197.1.8.1, local AS number 1
BGP table version is 72, main routing table version 72
41 network entries and 41 paths using 5453 bytes of memory
18 BGP path attribute entries using 936 bytes of memory
10 BGP AS-PATH entries using 256 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
5 BGP filter-list cache entries using 60 bytes of memory
16 received paths for inbound soft reconfiguration
BGP activity 300/2803 prefixes, 522/469 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.20.20   4    20    1504    1451       72    0    0 00:26:44        10
192.168.20.22   4    22    1123    1073        0    0    0 00:00:37 Idle (PfxCt)

WG1R1#show ip bgp neighbor 192.168.20.22
BGP neighbor is 192.168.20.22,  remote AS 22, external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read 00:04:48, hold time is 180, keepalive interval is 60 seconds
  Received 1123 messages, 0 notifications, 0 in queue
  Sent 1073 messages, 1 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
  Default weight 200

 For address family: IPv4 Unicast
  BGP table version 72, neighbor version 0
  Index 2, Offset 0, Mask 0x4
  Inbound soft reconfiguration allowed
  Inbound path policy configured
```

```
      Outbound path policy configured
      Incoming update prefix filter list is FromCheap
      Incoming update AS path filter list is 2
      Outgoing update AS path filter list is 3, maximum limit 15
      Threshold for warning message 75%

      Connections established 6; dropped 6
      Last reset 00:04:48, due to Error during connection collision
      Peer had exceeded the max. no. of prefixes configured.
      Reduce the no. of prefix and clear ip bgp 192.168.20.22 to restore peering
      No active TCP connection
WG1R1#
```

- Increase the number of allowed prefixes to continue testing. Clear the neighbor session between your WG*x*R1 router and router "Cheap" afterward.

- Inspect the prefixes that your router is receiving from "Cheap" with the **show ip bgp neighbors** command. Be sure to use the **routes** keyword. Your display should resemble the following:

```
WG1R1#show ip bgp nei 192.168.20.22 routes
BGP table version is 249, local router ID is 197.1.8.1
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
*  10.0.0.0         192.168.20.233                        0 22 20 i
*  128.20.0.0       192.168.20.22           0             0 22 i
*  128.20.12.0/24   192.168.20.20                         0 22 20 i
*  128.22.0.0       192.168.20.22           0             0 22 i
*> 128.22.12.0/24   192.168.20.22           0             0 22 i
*> 128.26.0.0       192.168.20.22           0             0 22 26 i
*  128.37.0.0       192.168.20.22           0             0 22 26 42 37 i
*  128.42.0.0       192.168.20.22           0             0 22 26 42 i
*> 128.51.0.0       192.168.20.22           0             0 22 26 51 i
*  128.213.0.0      192.168.20.20                         0 22 20 213 i
*> 128.214.0.0      192.168.20.22           0             0 22 214 i
*  192.20.11.0      192.168.20.22           0             0 22 i
*  192.20.12.0/30   192.168.20.20                         0 22 20 i
*  192.22.11.0      192.168.20.22           0             0 22 i
*> 192.22.12.0/30   192.168.20.22           0             0 22 i
*> 192.26.11.0      192.168.20.22           0             0 22 26 i
*  192.37.11.0      192.168.20.22           0             0 22 26 42 37 i
*  192.42.11.0      192.168.20.22           0             0 22 26 42 i
*> 192.51.11.0      192.168.20.22           0             0 22 26 51 i
*  192.213.11.0     192.168.20.20                         0 22 20 213 i
*> 192.214.11.0     192.168.20.22           0             0 22 214 i
*  200.20.0.0/16    192.168.20.20                         0 22 20 i
*> 200.22.0.0/16    192.168.20.22           0             0 22 i

Total number of prefixes 23
```

- Increase the number of allowed prefixes so that all of the prefixes coming from "Cheap" will be allowed and the session will not be terminated.

- Clear the neighbor session (configuration done in this exercise) between your WG*x*R1 router and router "Cheap" afterward and ensure that the neighbor session remains active.

Answer this question:

Q1)   What can you configure on your router to allow the BGP session to restart automatically if the received number of prefixes exceeds the configured maximum?

Lab Exercise 7-2: Implementing BGP Peer Groups

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will use BGP peer groups to share common configuration parameters between multiple BGP peers. After completing this exercise, you will be able to meet these objectives:

- Configure BGP peer groups

- Monitor the operation of BGP peer groups

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
neighbor *peer-group-name* **peer-group**	Creates a BGP or multiprotocol BGP peer group
neighbor *ip-address* **peer-group** *peer-group-name*	Configures a BGP neighbor to be a member of a peer group
neighbor *peer-group-name* **maximum-prefix** *maximum* [**threshold**] [**warning-only**][**restart** *restart-interval*]	Specifies the allowed maximum number of prefixes
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp neighbors *neighbor*	Shows detailed information about a specific neighbor
show ip bgp peer-group [*peer-group-name*] [**summary**]	Enables you to display information about BGP peer groups

Job Aids

These job aids are available to help you complete the lab exercise:

- After working with the "Cheap" provider, you have finally resolved the maximum-prefix violation problem that your network was experiencing. You have decided to increase the number of prefixes that are allowed and to make prefix limiting a universally applied policy for all peers.

- In this exercise, you will configure router WG*x*R1 to terminate the neighbor session between you and your upstream providers if you receive more than 50 BGP prefixes from them.

- You will use BGP peer groups to implement the policy to ease administration when more peers are added and to minimize router CPU utilization.

- Figure 1 shows the connectivity that is established between your AS and the two service providers "Good" and "Cheap."

Connecting to Two Different Service Providers

Cisco.com

Client

Client ISP

WGxR4

WGxR3

WGxR2

WGxR1

BGP session

BGP session

Good

Cheap

© 2004 Cisco Systems, Inc. All rights reserved.

BGP v3.1—6
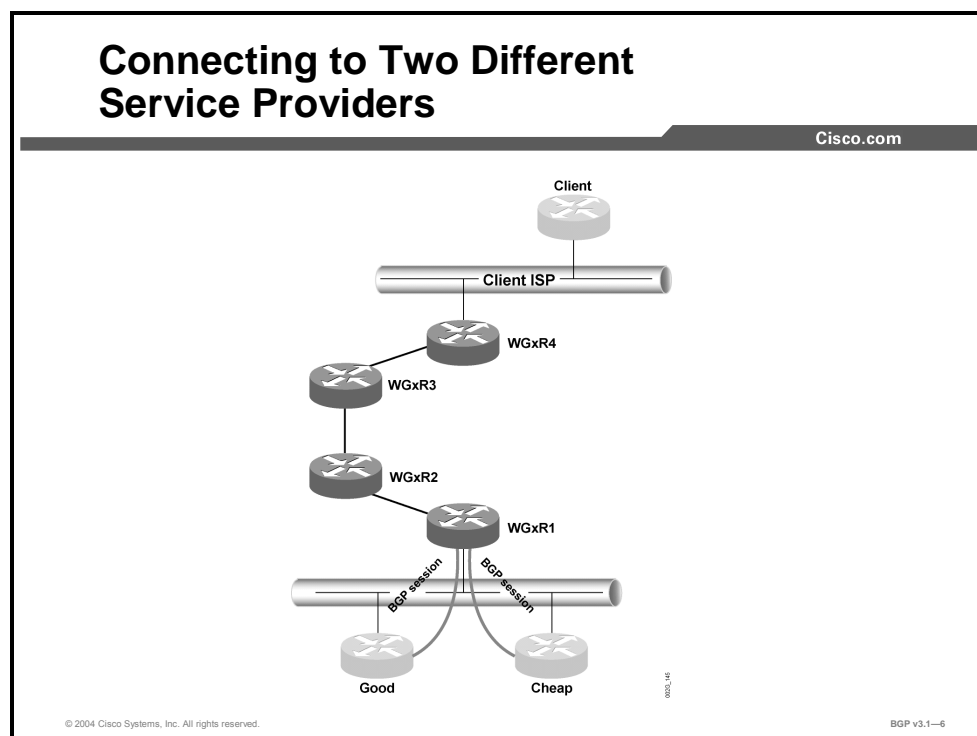
Figure 1: Connecting to Two Different Service Providers

Exercise Procedure

Complete these steps:

Creating a BGP peer group:

Step 1   Create a BGP peer group on your WG*x*R1 router to set a limit to the number of prefixes that your router accepts. BGP sessions should be terminated if the number of received prefixes exceeds 50.

Assigning a neighbor into a peer group:

Step 2   Configure the neighbor connections for the service provider routers "Good" and "Cheap" to be members of the peer group.

Exercise Verification

You have completed this exercise when you attain these results:

■ On WG*x*R1, use the **show ip bgp peer-group** *peer-group-name* command with the **summary** keyword to verify that you have configured the correct neighbors as members of the peer group. Your display should resemble the following:

```
WG1R1#show ip bgp peer-group SvcPro summary
BGP router identifier 197.1.8.1, local AS number 65001
BGP table version is 355, main routing table version 355
43 network entries and 66 paths using 6547 bytes of memory
20 BGP path attribute entries using 1040 bytes of memory
19 BGP AS-PATH entries using 456 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 activity 84/1135 prefixes, 319/253 paths, scan interval 15 secs
Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.20.20   4    20   13979    8418      355    0    0 00:15:29       23
192.168.20.22   4    22     412     397      355    0    0 00:07:38       23
```

■ On WG*x*R1, use the **show ip bgp peer-group** command to verify that the parameters of the peer group have been properly configured. Your display should resemble the following:

```
WG1R1#show ip bgp peer-group
BGP peer-group is SvcPro
  BGP version 4
  Default minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP neighbor is SvcPro, peer-group external, members:
     192.168.20.20   192.168.20.22
  Index 1, Offset 0, Mask 0x2
  Inbound soft reconfiguration allowed
  Update messages formatted 38, replicated 13
```

- Using the **show ip bgp neighbors** command, inspect the prefixes that your router is receiving from both "Good" and "Cheap." Be sure to use the **received-routes** keyword.

```
wg1r1#show ip bgp neighbors 192.168.20.20 received-routes
BGP table version is 355, local router ID is 197.1.8.1
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
*> 10.0.0.0         192.168.20.233         0               0 20 i
*> 128.20.0.0       192.168.20.20          0               0 20 i
*> 128.20.12.0/24   192.168.20.20          0               0 20 i
*> 128.22.0.0       192.168.20.20          0               0 20 i
*  128.22.12.0/24   192.168.20.22                          0 20 22 i
*  128.26.0.0       192.168.20.20          0               0 20 42 26 i
*> 128.37.0.0       192.168.20.20          0               0 20 42 37 i
*> 128.42.0.0       192.168.20.20          0               0 20 42 i
*  128.51.0.0       192.168.20.20          0               0 20 42 26 51 i
*> 128.213.0.0      192.168.20.20          0               0 20 213 i
*  128.214.0.0      192.168.20.22                          0 20 22 214 i
*> 192.20.11.0      192.168.20.20          0               0 20 i
*> 192.20.12.0/30   192.168.20.20          0               0 20 i
*> 192.22.11.0      192.168.20.20          0               0 20 i
*  192.22.12.0/30   192.168.20.22                          0 20 22 i
*  192.26.11.0      192.168.20.20          0               0 20 42 26 i
*> 192.37.11.0      192.168.20.20          0               0 20 42 37 i
*> 192.42.11.0      192.168.20.20          0               0 20 42 i
*  192.51.11.0      192.168.20.20          0               0 20 42 26 51 i
*> 192.213.11.0     192.168.20.20          0               0 20 213 i
*  192.214.11.0     192.168.20.22                          0 20 22 214 i
*> 200.20.0.0/16    192.168.20.20          0               0 20 i
*  200.22.0.0/16    192.168.20.22                          0 20 22 i
Total number of prefixes 23
```

- Clear the neighbor session between your WG*x*R1 router and router "Cheap" afterward and ensure that the neighbor session remains active.

Answer this question:

Q1)   If the number of prefixes that are received from both providers exceeds 50, what single command can you use to reset the BGP sessions on both routers?

Lab Exercise 7-3: Using BGP Route Dampening

Complete this lab exercise to practice what you learned in the related lesson.

Exercise Objective

In this exercise, you will use route dampening to minimize the impact of unstable routes. After completing this exercise, you will be able to meet these objectives:

- Configure BGP route dampening

- Release dampened routes

- Monitor BGP route dampening

Required Resources

These are the resources and equipment required to complete this exercise:

Your workgroup requires the following components:

- Four Cisco 2610 routers with a WIC-1T and BGP-capable operating system software installed.

- Four CAB-X21FC + CAB-X21MT DTE-DCE serial cable combinations. The DCE side of the cable is connected to the Cisco 3660.

- Two Ethernet 10BASE-T patch cables.

- IBM PC (or compatible) with Windows 95/98 and an installed Ethernet adapter.

The lab backbone requires the following components (supporting up to eight workgroups):

- One Cisco 2610 router with a WIC-1T and BGP-capable operating system software installed

- Two Cisco 2610 routers with BGP-capable operating system software installed

- One Cisco 3640 router with an installed NM-8A/S

- Two Catalyst 2924M-XL Ethernet switches

- Three Ethernet 10BASE-T patch cables

Command List

The commands used in this exercise are described in the table here.

Commands

Command	Description
router bgp *as-number*	Places you in BGP configuration mode
bgp dampening route-map *route-map*	Specifies BGP dampening parameters by using a route-map
match community *c-acl*	Matches BGP communities in a route-map by using a community-list
set dampening *half-life reuse suppress max-suppress-time*	Sets the BGP route-dampening factors
ip community-list *num* {**permit** | **deny**} *community*	Creates a community access-list
show ip bgp	Enables you to inspect the contents of the BGP table
show ip bgp flap-statistics	Enables you to view prefixes that have previously flapped
show ip bgp dampened-paths	Enables you to view prefixes that are currently suppressed
show ip bgp regexp *regexp*	Enables you to use a regular expression to filter the output of the **show ip bgp** command
show ip bgp community [*community* [*community …*]] [**exact-match**]	Enables you to view BGP routes that have at least one BGP community attribute or those that are specified in the command
show ip bgp community-list *c-list* [**exact-match**]	Enables you to view BGP routes that are permitted by the specified community-list
debug ip bgp dampening	Enables you to see major events that are related to route dampening

Job Aids

These job aids are available to help you complete the lab exercise:

- Service providers usually use BGP route dampening to minimize the impact that route flaps in the Internet can have on the stability of their networks. BGP route dampening also reduces the CPU load on the routers that are running BGP. In this exercise, you will minimize the impact of flapping routes on your AS and upstream autonomous systems by implementing BGP route dampening.

- Configure graded route dampening for routes that are received from the "Client" router according to the following design:

    — If a prefix contains a community attribute of x:300, then the prefix should not be dampened.

    — Prefixes longer than /23 should have a half-life of 20 min.

    — Prefixes between /9 and /22 should have a half-life of 15 min.

    — Prefixes between /0 and /8 should have a half-life of 10 min.

    — Maximum suppression time should be changed to four times the half-life time for all prefixes.

- Figure 1 shows the connectivity that is established between your AS and the "Client" router.
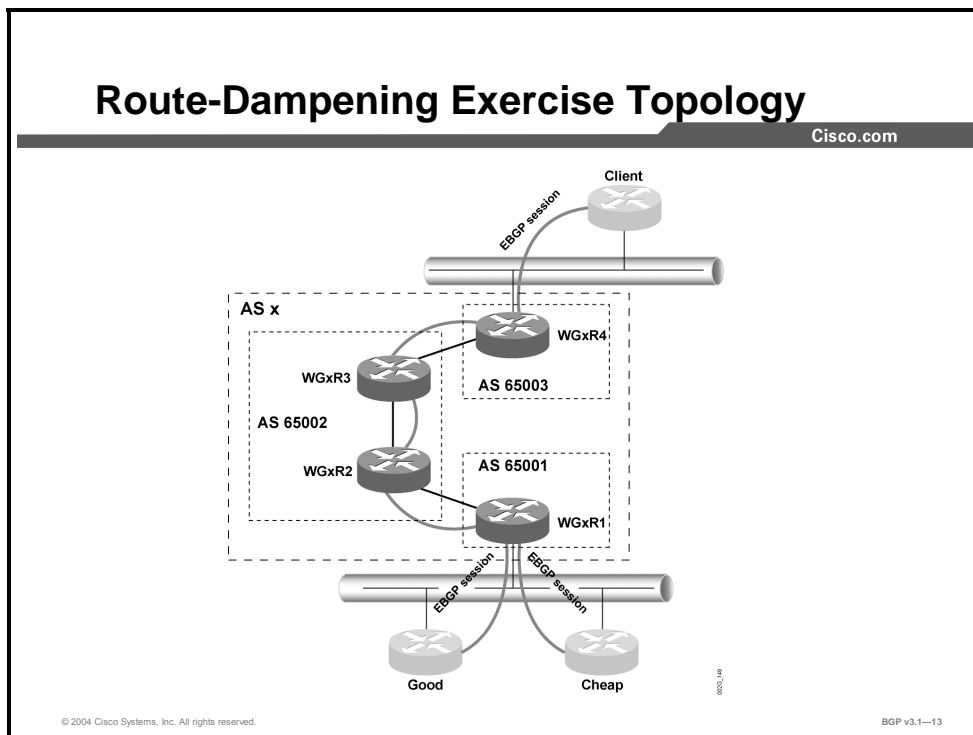
Figure 1: Route-Dampening Exercise Topology

Exercise Procedure

Complete these steps:

Creating filters for the dampening route-map:

Step 1   Create a community-list that matches prefixes that carry a BGP community of *x*:300 (where *x* is your real AS number, not your member-AS number).

Step 2   Create a prefix-list that matches all prefixes that are longer than /23.

Step 3   Create a prefix-list that matches all prefixes that are longer than /8.

Creating a route-map for route dampening:

Step 4   Create a new route-map. The first statement of the new route-map should match prefixes that carry a BGP community of *x*:300. No route-dampening parameters need to be set.

Step 5   The second statement should match prefixes that are longer than /23. Set the *half-life* parameter to 20 min and *max-suppress-time* to 80 min, and use default values for other parameters (reuse 750, suppress 2000).

Step 6   The third statement should match prefixes that are longer than /8. Set the *half-life* parameter to 15 min and *max-suppress-time* to 60 min, and use default values for other parameters (reuse 750, suppress 2000).

Step 7   The last statement should match all other prefixes (**no match** command). Set the *half-life* parameter to 10 min and *max-suppress-time* to 40 min, and use default values for other parameters (reuse 750, suppress 2000).

Step 8   Enter BGP configuration mode and, using the **bgp dampening route-map** *route-map* command, start BGP route dampening based on the created route-map.

Exercise Verification

You have completed this exercise when you attain these results:

■ Shut down interface Loopback55 on router "Client" several times to create flaps. Use **show ip bgp flap-statistics** and **show ip bgp dampened-paths** to verify the operation of route dampening on your WG*x*R4 router. Your display should resemble the following:

```
WGxR4#show ip bgp damp
BGP table version is 89, local router ID is 197.1.7.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From            Reuse     Path
*d 99.0.0.0         192.168.21.99    00:25:20 99 i
*d 197.99.128.0/20  192.168.21.99    00:40:30 99 i


WGxR4#show ip bgp flap
BGP table version is 89, local router ID is 197.1.7.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          From           Flaps Duration Reuse     Path
*d 99.0.0.0         192.168.21.99   6     00:09:25 00:25:20 99
*d 197.99.128.0/20  192.168.21.99   6     00:09:25 00:40:20 99
```

```
WGxR4#show ip bgp 99.0.0.0
BGP routing table entry for 99.0.0.0/8, version 80
Paths: (1 available, no best path)
  Not advertised to any peer
  99, (suppressed due to dampening)
    192.168.21.99 from 192.168.21.99 (197.99.111.1)
      Origin IGP, metric 0, localpref 100, valid, external
      Dampinfo: penalty 958, flapped 6 times in 00:31:20, reuse in 00:03:30


WGxR4#show ip bgp
BGP table version is 89, local router ID is 197.1.7.1
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
*> 10.0.0.0         192.168.20.233        0    100      0 (65002 65001) 20 i
*d 99.0.0.0         192.168.21.99         0             0 99 i
*> 128.20.0.0       192.168.20.20         0    100      0 (65002 65001) 20 i
*> 128.20.12.0/24   192.168.20.20         0    100      0 (65002 65001) 20 i
*> 128.22.0.0       192.168.20.20         0    100      0 (65002 65001) 20 i
*> 128.22.12.0/24   192.168.20.22         0    100      0 (65002 65001) 22 i
*> 128.26.0.0       192.168.20.22         0    100      0 (65002 65001) 22 26 i
*> 128.37.0.0       192.168.20.20         0    100      0 (65002 65001) 20 42 37 i
*> 128.42.0.0       192.168.20.20         0    100      0 (65002 65001) 20 42 i
*> 128.51.0.0       192.168.20.22         0    100      0 (65002 65001) 22 26 51 i
*> 128.213.0.0      192.168.20.20         0    100      0 (65002 65001) 20 213 i
*> 128.214.0.0      192.168.20.22         0    100      0 (65002 65001) 22 214 i
*> 192.20.11.0      192.168.20.20         0    100      0 (65002 65001) 20 i
*> 192.20.12.0/30   192.168.20.20         0    100      0 (65002 65001) 20 i
*> 192.22.11.0      192.168.20.20         0    100      0 (65002 65001) 20 i
*> 192.22.12.0/30   192.168.20.22         0    100      0 (65002 65001) 22 i
*> 192.26.11.0      192.168.20.22         0    100      0 (65002 65001) 22 26 i
*> 192.37.11.0      192.168.20.20         0    100      0 (65002 65001) 20 42 37 i
*> 192.42.11.0      192.168.20.20         0    100      0 (65002 65001) 20 42 i
*> 192.51.11.0      192.168.20.22         0    100      0 (65002 65001) 22 26 51 i
*> 192.168.1.0      0.0.0.0               0         32768 i
*> 192.213.11.0     192.168.20.20         0    100      0 (65002 65001) 20 213 i
*> 192.214.11.0     192.168.20.22         0    100      0 (65002 65001) 22 214 i
*> 197.1.1.0        197.1.1.1             0    100      0 (65002 65001) i
*> 197.1.2.0        197.1.2.1             0    100      0 (65002) i
*> 197.1.3.0        197.1.2.1             0    100      0 (65002) i
*> 197.1.4.0        197.1.4.1             0    100      0 (65002) i
*> 197.1.5.0        197.1.4.1             0    100      0 (65002) i
*> 197.1.6.0        0.0.0.0               0         32768 i
*> 197.1.7.0        0.0.0.0               0         32768 i
*> 197.1.8.0        197.1.1.1             0    100      0 (65002 65001) i
*> 197.99.1.0       192.168.21.99         0             0 99 i
*> 197.99.11.0      192.168.21.99         0             0 99 i
*> 197.99.12.0      192.168.21.99         0             0 99 i
*> 197.99.13.0      192.168.21.99         0             0 99 i
*> 197.99.20.0      192.168.21.99         0             0 99 i
*> 197.99.22.0      192.168.21.99         0             0 99 i
*> 197.99.80.0      192.168.21.99         0             0 99 i
*> 197.99.111.1/32  192.168.21.99         0             0 99 i
*> 197.99.120.0     192.168.21.99         0             0 99 i
*d 197.99.128.0/20  192.168.21.99         0             0 99 i
*> 200.20.0.0/16    192.168.20.20         0    100      0 (65002 65001) 20 i
*> 200.22.0.0/16    192.168.20.22         0    100      0 (65002 65001) 22 i
```

Answer these questions:

Q1)   What is the purpose of route dampening?

Q2)   Which routes are affected by route dampening?

Lab Exercise Answer Key

Lab Exercise 1-1: Initial Lab Setup

There are no specific solutions for the "Initial Lab Setup."

Lab Exercise 1-2: Configuring Basic BGP

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Task 1: Configuring BGP

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1#configure terminal
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.20 remote-as 20
WGxR1(config-router)#network 192.168.x.0
WGxR1(config-router)#network 197.x.0.0 mask 255.255.0.0
WGxR1(config-router)#exit
WGxR1(config)#ip route 197.x.0.0 255.255.0.0 null 0 250
WGxR1(config)#end
```

Also, make sure that you announce a default route into your network. If you are running OSPF in your network, the command sequence to use is as follows:

```
WGxR1#configure terminal
WGxR1(config)#router ospf process-id
WGxR1(config-router)#default-information originate always
```

Answers to Review Questions:

Q1)   What do you need in order to be able to propagate classful networks?

There must be at least one subnet in the routing table, and the network must be listed in the BGP process with no mask attached to it.

Q2)   What do you need to be able to propagate classless networks (supernets or subnets)?

The prefix must be listed in the BGP process together with its proper mask. The exact corresponding entry must also be present in the IP routing table.

Q3)   Why do some networks that are received from router "Good" have a next-hop address pointing to other routers?

Router "Good" performs next-hop processing and sets the next hop to router "Cheap" for routes that are received from "Cheap" and propagated to you.

Q4)   What command would you use to see if a neighbor is sending you any updates and how many?

You would use the **show ip bgp neighbor** command.

Task 2: Configuring Route Redistribution in BGP

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1#configure terminal
WGxR1(config)#router bgp x
WGxR1(config-router)#no network 192.168.x.0
WGxR1(config-router)#no network 197.x.0.0 mask 255.255.0.0
WGxR1(config-router)#redistribute <igp> <pid> route-map SetOrigin
WGxR1(config)#access-list 1 deny   192.168.20.0
WGxR1(config)#access-list 1 deny   192.168.21.0
WGxR1(config)#access-list 1 permit any
WGxR1(config)#route-map SetOrigin permit 10
WGxR1(config-route-map)#match ip address 1
WGxR1(config-route-map)#set origin igp
WGxR1(config-route-map)#end
WGxR1#
```

Answers to Review Questions:

Q1)   What is the major difference between this implementation and the previous one? Which is better and why?

The previous implementation is better for service provider environments because it is more controlled. This implementation is better for enterprise networks because it is more dynamic.

Q2)   What precautions do you have to take when using redistribution?

You should always filter redistributed routes with a route-map to make sure that unwanted networks are not announced into BGP.

Task 3: Configuring BGP Aggregation

Use the following command sequence on your router WGxR1 (replace x with your workgroup number):

```
WGxR1(config)#router bgp x
WGxR1(config-router)#aggregate-address 197.x.0.0 255.255.0.0 summary-only
WGxR1(config-router)#aggregate-address 197.x.8.0 255.255.252.0 summary-only
WGxR1(config-router)#end
WGxR1#
```

Answers to Review Questions:

Q1)   Do you see all your prefixes on the provider router? Why?

You should see only the configured aggregates and the routes that are not within the aggregation range on the provider router. All the other routes should be suppressed.

Q2)   What do you need to be able to generate and propagate aggregates?

The aggregate address has to be configured in the BGP process, and a prefix within the aggregate range has to be present in the BGP table.

Before Aggregation:

```
> 197.1.1.0          192.168.20.1          0          0 1 i
> 197.1.2.0          192.168.20.1          0          0 1 i
> 197.1.3.0          192.168.20.1          0          0 1 i
> 197.1.4.0          192.168.20.1          0          0 1 i
> 197.1.5.0          192.168.20.1          0          0 1 i
> 197.1.6.0          192.168.20.1          0          0 1 i
> 197.1.7.0          192.168.20.1          0          0 1 i
> 197.1.8.0          192.168.20.1          0          0 1 I
```

After Aggregation:

```
*> 197.1.0.0/16     192.168.20.1                     0 1 i
*> 197.1.8.0/22     192.168.20.1                     0 1 I
```

Lab Exercise 2-1: Configuring a Transit AS

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Task 1: Configuring the BGP Transit AS

Use the following command sequence on your router WG*x*R1:

```
WGxR1(config)#router bgp x
WGxR1(config-router)#no synchronization
WGxR1(config-router)#neighbor 192.168.20.20 remote-as 20
WGxR1(config-router)#neighbor 197.x.2.1 remote-as x
WGxR1(config-router)#neighbor 197.x.2.1 update-source loopback 0
WGxR1(config-router)#neighbor 197.x.2.1 next-hop-self
WGxR1(config-router)#neighbor 197.x.4.1 remote-as x
WGxR1(config-router)#neighbor 197.x.4.1 update-source loopback 0
WGxR1(config-router)#neighbor 197.x.4.1 next-hop-self
WGxR1(config-router)#neighbor 197.x.6.1 remote-as x
WGxR1(config-router)#neighbor 197.x.6.1 update-source Loopback 0
WGxR1(config-router)#neighbor 197.x.6.1 next-hop-self
WGxR1(config-router)#network 192.168.1.0
WGxR1(config-router)#network 197.1.0.0 mask 255.255.0.0
WGxR1(config-router)#end
```

Use the following command sequence on your router WG*x*R2:

```
WGxR2(config)#router bgp x
WGxR2(config-router)#no synchronization
WGxR2(config-router)#neighbor 197.x.1.1 remote-as x
WGxR2(config-router)#neighbor 197.x.1.1 update-source Loopback0
```

```
WGxR2(config-router)#neighbor 197.x.4.1 remote-as x
WGxR2(config-router)#neighbor 197.x.4.1 update-source Loopback0
WGxR2(config-router)#neighbor 197.x.6.1 remote-as x
WGxR2(config-router)#neighbor 197.x.6.1 update-source Loopback0
WGxR2(config-router)#end
```

Use the following command sequence on your router WG*x*R3:

```
WGxR3(config)#router bgp x
WGxR3(config-router)#no synchronization
WGxR3(config-router)#neighbor 197.x.1.1 remote-as x
WGxR3(config-router)#neighbor 197.x.1.1 update-source Loopback0
WGxR3(config-router)#neighbor 197.x.2.1 remote-as x
WGxR3(config-router)#neighbor 197.x.2.1 update-source Loopback0
WGxR3(config-router)#neighbor 197.x.6.1 remote-as x
WGxR3(config-router)#neighbor 197.x.6.1 update-source Loopback0
WGxR3(config-router)#end
```

Use the following command sequence on your router WG*x*R4:

```
WGxR4(config)#router bgp x
WGxR4(config-router)#no synchronization
WGxR4(config-router)#neighbor 192.168.21.99 remote-as 99
WGxR4(config-router)#network 192.168.1.0
WGxR4(config-router)#network 197.1.0.0 mask 255.255.0.0
WGxR4(config-router)#neighbor 197.x.1.1 remote-as x
WGxR4(config-router)#neighbor 197.x.1.1 update-source Loopback 0
WGxR4(config-router)#neighbor 197.x.1.1 next-hop-self
WGxR4(config-router)#neighbor 197.x.2.1 remote-as x
WGxR4(config-router)#neighbor 197.x.2.1 update-source loopback 0
WGxR4(config-router)#neighbor 197.x.2.1 next-hop-self
WGxR4(config-router)#neighbor 197.x.4.1 remote-as x
WGxR4(config-router)#neighbor 197.x.4.1 update-source loopback 0
WGxR4(config-router)#neighbor 197.x.4.1 next-hop-self
WGxR4(config-router)#exit
WGxR4(config)# ip route 197.x.0.0 255.255.0.0 null 0 250
WGxR4(config-)#end
```

Answers to Review Questions:

Q1)   Check the BGP table on router "Client." How many prefixes coming from your AS are in that BGP table?

20

Q2)   Is there any other way of discovering how many prefixes that you have advertised to the router "Client"?

show ip bgp neighbor *ip-address* **advertised**

Q3)   Why did you have to disable synchronization?

You should use BGP synchronization only in networks where BGP routes are redistributed in IGP. This design is no longer viable for service provider networks because of the large number of routes that are advertised in the Internet.

Q4)   Why did you have to establish a full mesh of IBGP sessions?

IBGP split-horizon rules require a full mesh of IBGP sessions within the AS.

Lab Exercise 3-1: Using Multihomed BGP Networks

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.22 remote-as 22
WGxR1(config-router)#neighbor 192.168.20.22 weight 100
WGxR1(config-router)#end
WGxR1#
```

Answers to Review Questions:

Q1) What can happen if a multihomed AS is passing routing information, learned from one neighbor, to another neighbor?

The multihomed AS that is passing routes between BGP neighbors can become a transit AS.

Q2) Why do some prefixes have two paths but both use the same next-hop address?

Both service providers are connected to the same subnet as all the customers, and the BGP next-hop processing optimizes the data flow.

Lab Exercise 3-2: Employing AS-Path Filters

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1(config)#ip as-path access-list 1 deny _214_
WGxR1(config)#ip as-path access-list 1 permit .*
WGxR1(config)#ip as-path access-list 2 deny _213$
WGxR1(config)#ip as-path access-list 2 permit .*
WGxR1(config)#ip as-path access-list 3 permit ^$
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.20 filter-list 1 in
WGxR1(config-router)#neighbor 192.168.20.22 filter-list 2 in
WGxR1(config-router)#neighbor 192.168.20.20 filter-list 3 out
WGxR1(config-router)#neighbor 192.168.20.22 filter-list 3 out
WGxR1(config-router)#end
WGxR1#clear ip bgp *
```

Answers to Review Questions:

Q1)   By this time, the BGP table should contain a large number of prefixes. What regular expression would you use with the **show ip bgp command** on WG*x*R1 to view the prefixes that are originated by your AS?

Use **show ip bgp regexp ^$**.

```
wg1r1#show ip bgp regexp ^$
BGP table version is 44, local router ID is 197.1.8.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - inter
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.1.0      0.0.0.0                  0          32768 i
*> 197.1.0.0/16     0.0.0.0                             32768 i
```

Q2)   What regular expression would you use on router "Good" or "Cheap" to view the prefixes that originate in your AS?

Use **show ip bgp regexp _x$.**

AS is 1.

```
Good# show ip bgp regexp _1$
BGP table version is 111, local router ID is 199.199.199.199
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
*  192.168.1.0      192.168.20.1                        0 22 1 i
*>                  192.168.20.1             0            0 1 i
*  197.1.0.0/16     192.168.20.1                        0 22 1 i
*>                  192.168.20.1                         0 1 i
```

Q3)   What regular expression would you use on router "Good" or "Cheap" to view the prefixes that are received from your AS?

Use **show ip bgp regexp ^x_.**

```
Good#show ip bgp regexp ^1_
BGP table version is 111, local router ID is 199.199.199.199
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
*> 192.168.1.0      192.168.20.1             0            0 1 i
*> 197.1.0.0/16     192.168.20.1                         0 1 i
```

Q4)   How can you test your AS-path filters before applying them and clearing BGP neighbors?

You can test the AS-path filters with the **show ip bgp filter-list** *filter* command.

Lab Exercise 3-3: Filtering with Prefix-Lists

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1(config)#ip prefix-list FromGood deny 192.0.0.0/3 ge 24
WGxR1(config)#ip prefix-list FromGood deny 192.168.0.0/16 le 32
WGxR1(config)#ip prefix-list FromGood deny 172.16.0.0/12 le 32
WGxR1(config)#ip prefix-list FromGood deny 10.0.0.0/8 le 32
WGxR1(config)#ip prefix-list FromGood permit 0.0.0.0/0 le 32
WGxR1(config)#ip prefix-list FromCheap deny 128.0.0.0/2 ge 17
WGxR1(config)#ip prefix-list FromCheap deny 192.168.0.0/16 le 32
WGxR1(config)#ip prefix-list FromCheap deny 172.16.0.0/12 le 32
WGxR1(config)#ip prefix-list FromCheap deny 10.0.0.0/8 le 32
WGxR1(config)#ip prefix-list FromCheap permit 0.0.0.0/0 le 32
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.20 prefix-list FromGood in
WGxR1(config-router)#neighbor 192.168.20.22 prefix-list FromCheap in
WGxR1(config-router)#end
WGxR1#clear ip bgp *
```

Answers to Review Questions:

Q1)   Are neighboring workgroups still reachable?

Yes, they are still reachable.

Q2)   Why does router WG*x*R1 still accept class B networks from router "Cheap"?

Only subnets of class B networks are filtered, not the class B networks themselves.

Lab Exercise 3-4: Implementing Changes in BGP Policy

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
wg1r1#show ip bgp neighbor 192.168.20.20 received
% Inbound soft reconfiguration not enabled

wg1r1#show ip bgp neighbor 192.168.20.22 received
% Inbound soft reconfiguration not enabled

wg1r1(config)#router bgp x
wg1r1(config-router)#neighbor 192.168.20.20 soft-reconfiguration inbound
wg1r1(config-router)#neighbor 192.168.20.22 soft-reconfiguration inbound
wg1r1(config-router)#exit
wg1r1(config)#exit
```

Answers to Review Questions:

> Q1) What command do you use to show which entries in the BGP table of the local router have been propagated to a specific neighbor?
>
> **show ip bgp neighbors** *ip-address* **advertised**

> Q2) What command do you use to display which of the routes in the local BGP table have been received and accepted from an indicated neighbor?
>
> **show ip bgp neighbors** *ip-address* **routes**

Lab Exercise 4-1: Influencing BGP Route Selection with Weights

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

```
WGxR1(config)#ip as-path access-list 10 permit _37_|_213_
WGxR1(config)#route-map SetWeight permit 10
WGxR1(config-route-map)#match as-path 10
WGxR1(config-route-map)#set weight 300
WGxR1(config-route-map)#exit
WGxR1(config)#route-map SetWeight permit 20
WGxR1(config-route-map)#exit
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.20 weight 100
WGxR1(config-router)#neighbor 192.168.20.22 weight 200
WGxR1(config-router)#neighbor 192.168.20.20 route-map SetWeight in
WGxR1(config-router)#end
WGxR1#clear ip bgp *
WGxR1#
```

Answers to Review Questions:

> Q1) Did all paths automatically get a weight of 100 or 200? Why not? What did you have to do?
>
> The BGP sessions had to be reset in order to apply newly configured weights to incoming BGP routes.

> Q2) Name some parameters and attributes that are used for best-path selection.
>
> The following BGP parameters and attributes are used for BGP best-path selection in the order specified: weight, local preference, AS-path length, origin code, MED.

Lab Exercise 4-2: Setting BGP Local Preference

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your workgroup routers as indicated (replace *x* with your workgroup number):

Configuration of router WG*x*R2:

```
WGxR2(config)#route-map SetLP
WGxR2(config-route-map)#set local-preference 200
WGxR2(config-route-map)#exit
WGxR2(config)#router bgp x
WGxR2(config)#router# no synchronization
WGxR2(config-router)#network 192.168.x.0
WGxR2(config-router)#neighbor 192.168.3x.2 remote-as 20
WGxR2(config-router)#neighbor 192.168.3x.2 route-map SetLP in
WGxR2(config-router)#exit
WGxR2(config)#int s0/0.200 point-to-point
WGxR2(config-subif)#ip address 192.168.31.1 255.255.255.252
WGxR2(config-subif)#frame-relay interface-dlci 20x
WGxR2(config-fr-dlci)#exit
WGxR2(config-subif)#exit
WGxR2(config)#exit
```

Configuration of router WG*x*R1:

```
WGxR1(config)router bgp x
WGxR1(config-router)#no synchronization
WGxR1(config-router)#no neighbor 192.168.20.20 weight 100
WGxR1(config-router)#no neighbor 192.168.20.22 weight 200
WGxR1(config-router)#no neighbor 192.168.20.20 route-map SetWeight in
WGxR1(config-router)#end
```

Answers to Review Questions

Q1)   Is routing between router WG*x*R1 and "Good" symmetrical?

Initially, the routing is not symmetrical, because the router "Good" prefers more stable EBGP routes received from WG*x*R1 and the router WG*x*R1 prefers routes received from WG*x*R2.

Q2)   Which routers receive the local preference attribute?

Only routers within the AS receive the local preference attribute.

Lab Exercise 4-3: Understanding BGP Multi-Exit Discriminators

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1 (replace *x* with your workgroup number):

Configuration of router WG*x*R1:

```
WGxR1(config)#route-map SetMED
WGxR1(config-route-map)#set metric 200
WGxR1(config-route-map)#exit
WGxR1(config)#router bgp x
WGxR1(config-router)#neighbor 192.168.20.20 route-map SetMED out
WGxR1(config-router)#end
WGxR1#
WGxR1#clear ip bgp *
```

Configuration of router WG*x*R2:

```
WGxR2(config)#route-map SetMED
WGxR2(config-route-map)#set metric 100
WGxR2(config-route-map)#exit
WGxR2(config)#router bgp x
WGxR2(config-router)#neighbor 192.168.3x.2 route-map SetMED out
WGxR2(config-router)#end
WGxR2#
WGxR2#clear ip bgp *
```

Answers to Review Questions

Q1)   Which parameters and attributes have to be equal before the MED is compared to select the best path?

The MED is used only to compare routes that are received from the same AS that have equal weight, local preference, AS-path length, and origin code.

Q2)   What is the default value of the MED?

For networks that originated in the local AS, the default MED value is the IGP metric from the IP routing table. For transit networks, the MED is removed by the router in outgoing EBGP updates.

Lab Exercise 4-4: Addressing BGP Communities

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R2 (replace *x* with your workgroup number):

Configuration of router WG*x*R2:

```
WG1R2(config)#route-map SetCommunity
WG1R2(config-route-map)#set community 1:20
WG1R2(config-route-map)#exit
WG1R2(config)#router bgp x
WG1R2(config-router)#neighbor 192.168.3x.2 route-map SetCommunity out
WG1R2(config-router)#neighbor 192.168.3x.2 send-community
WG1R(config-router)#exit
WG1R2(config)#ip bgp-community new-format
WG1R2(config)#exit
WG1R2#clear ip bgp *
```

Answers to Review Questions

Q1)   What do you have to do to enable community propagation?

Enable BGP community propagation with the **neighbor send-community** command.

Q2)   What mechanisms can you use to match or set communities?

Only route-maps allow matching or setting of BGP communities.

Lab Exercise 6-1: Introducing Route Reflectors

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1:

```
WGxR1(config)#router bgp x
WGxR1(config-router)#no neighbor 197.x.4.1 remote-as x
WGxR1(config-router)#no neighbor 197.x.6.1 remote-as x
WGxR1(config-router)#end
WGxR1#
```

Use the following command sequence on your router WG*x*R2:

```
WGxR2(config)#router bgp x
WGxR2(config-router)#bgp cluster-id 102
WGxR2(config-router)#no neighbor 197.x.6.1 remote-as x
WGxR2(config-router)#neighbor 197.x.1.1 route-reflector-client
WGxR2(config-router)#end
```

Use the following command sequence on your router WG*x*R3:

```
WGxR3(config)#router bgp x
WGxR3(config-router)#bgp cluster-id 101
WGxR3(config-router)#no neighbor 197.x.1.1 remote-as x
WGxR3(config-router)#neighbor 197.x.2.1 route-reflector-client
WGxR3(config-router)#end
```

Use the following command sequence on your router WG*x*R4:

```
WGxR4(config)#router bgp x
WGxR4(config-router)#no neighbor 197.x.2.1 remote-as x
WGxR4(config-router)#no neighbor 197.x.1.1 remote-as x
WGxR4(config-router)#end
```

Answers to Review Questions:

Q1)   Did this design require you to configure a cluster-ID?

This design did not require specified cluster-IDs, because there are no redundant route reflectors in the network.

Q2)   What is the default cluster-ID?

The default cluster-ID is the BGP router-ID.

Q3)   When do you have to configure a cluster-ID?

You need to configure a cluster-ID in redundant route reflector designs.

Lab Exercise 6-2: Configuring and Monitoring Confederations

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1:

```
WGxR1(config)#no router bgp x
WGxR1(config)#router bgp 65001
WGxR1(config-router)#network 192.168.x.0
WGxR1(config-router)#network 197.x.1.0
WGxR1(config-router)#network 197.x.8.0
WGxR1(config-router)#bgp confederation identifier x
WGxR1(config-router)#bgp confederation peers 65002
WGxR1(config-router)#neighbor 192.168.20.20 remote-as 20
WGxR1(config-router)#neighbor 192.168.20.22 remote-as 22
WGxR1(config-router)#neighbor 197.x.2.1 remote-as 65002
WGxR1(config-router)#neighbor 197.x.2.1 update-source loopback0
WGxR1(config-router)#neighbor 197.x.2.1 ebgp-multihop
WGxR1(config-router)#end
```

Use the following command sequence on your router WG*x*R2:

```
WGxR2(config)#no router bgp x
WGxR2(config)#router bgp 65002
WGxR2(config-router)#no synchronization
WGxR2(config-router)#network 197.x.2.0
WGxR3(config-router)#network 197.x.3.0
WGxR2(config-router)#bgp confederation identifier x
WGxR2(config-router)#bgp confederation peers 65001
WGxR2(config-router)#neighbor 197.x.1.1 remote-as 65001
WGxR2(config-router)#neighbor 197.x.1.1 update-source loopback0
WGxR2(config-router)#neighbor 197.x.1.1 ebgp-multihop
WGxR2(config-router)#neighbor 197.x.4.1 remote-as 65002
WGxR2(config-router)#neighbor 197.x.4.1 update-source loopback0
WGxR2(config-router)#end
```

Use the following command sequence on your router WG*x*R3:

```
WGxR3(config)#no router bgp x
WGxR3(config)#router bgp 65002
WGxR3(config-router)#no synchronization
WGxR3(config-router)#network 197.x.4.0
WGxR3(config-router)#network 197.x.5.0
WGxR3(config-router)#bgp confederation identifier x
WGxR3(config-router)#bgp confederation peers 65003
WGxR3(config-router)#neighbor 197.x.2.1 remote-as 65002
WGxR3(config-router)#neighbor 197.x.2.1 update-source loopback0
WGxR3(config-router)#neighbor 197.x.6.1 remote-as 65003
WGxR3(config-router)#neighbor 197.x.6.1 update-source loopback0
WGxR3(config-router)#neighbor 197.x.6.1 ebgp-multihop
WGxR3(config-router)#end
```

Use the following command sequence on your router WG*x*R4:

```
WGxR4(config)#no router bgp x
WGxR4(config)#ip as-path access-list 10 permit ^99$
WGxR4(config)#ip as-path access-list 11 permit ^$|\)$|_37$
WGxR4(config)#router bgp 65003
WGxR4(config-router)#network 192.168.x.0
WGxR4(config-router)#network 197.x.6.0
WGxR4(config-router)#network 197.x.7.0
WGxR4(config-router)#bgp confederation identifier x
WGxR4(config-router)#bgp confederation peers 65002
WGxR4(config-router)#neighbor 192.168.21.99 remote-as 99
WGxR4(config-router)#neighbor 192.168.21.99 filter-list 10 in
WGxR4(config-router)#neighbor 192.168.21.99 filter-list 11 out
WGxR4(config-router)#neighbor 197.x.4.1 remote-as 65002
WGxR4(config-router)#neighbor 197.x.4.1 update-source loopback0
WGxR4(config-router)#neighbor 197.x.4.1 ebgp-multihop
WGxR4(config-router)#end
```

Answers to Review Questions:

Q1)   What additional command did you have to use to establish intraconfederation EBGP sessions between loopback interfaces?

To establish intraconfederation EBGP sessions between loopback interfaces, you must specify **ebgp-multihop** on the intraconfederation EBGP neighbor.

Q2)   Why was it necessary to change the AS-path filters?

The AS path of the local network has changed, because it includes the member-AS numbers.

Lab Exercise 7-1: Limiting the Number of Prefixes Received from a BGP Neighbor

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1:

```
WGxR1(config)#router bgp 65001
WG1R1(config-router)#neighbor 192.168.20.22 maximum-prefix 15 warning-only
WG1R1#clear ip bgp 192.168.20.22

WG1R1(config)#router bgp 65001
WG1R1(config-router)#no neighbor 192.168.20.22 maximum-prefix 15 warning-only
WG1R1(config-router)#neighbor 192.168.20.22 maximum-prefix 15
WG1R1#clear ip bgp 192.168.20.22
```

Answers to Review Questions:

Q1)   What can you configure on your router to allow the BGP session to restart automatically if the received number of prefixes exceeds the configured maximum?

restart parameter of the **maximum-prefix** command:

   neighbor *ip-address* **maximum-prefix** *maximum* **restart**

Lab Exercise 7-2: Implementing BGP Peer Groups

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R1:

```
WG1R1(config)#router bgp 65001
WG1R1(config-router)#neighbor SvcPro peer-group
WG1R1(config-router)#neighbor 192.168.20.20 peer-group SvcPro
WG1R1(config-router)#neighbor 192.168.20.22 peer-group SvcPro
WG1R1(config-router)#neighbor SvcPro maximum-prefix 50
WG1R1(config-router)#neighbor SvcPro soft-reconfiguration in
WG1R1(config-router)#end
```

Answers to Review Questions:

Q1)   If the number of prefixes that are received from both providers exceeds 50, what single command can you use to reset the BGP sessions on both routers?

clear ip bgp peer-group *peer-group-name* **soft in**

In this example it would be the following:

clear ip bgp peer-group *SvcPro* **soft in**

Lab Exercise 7-3: Using BGP Route Dampening

When you complete this lab exercise, your router configuration will be similar to the following, with differences that are specific to your pod.

Use the following command sequence on your router WG*x*R4:

```
WGxR4(config)#ip community-list 10 permit x:300
WGxR4(config)#ip prefix-list RFD24 permit 0.0.0.0/0 ge 24
WGxR4(config)#ip prefix-list RFD8 permit 0.0.0.0/0 ge 9
WGxR4(config)#route-map RFD permit 10
WGxR4(config-route-map)#match community 10
WGxR4(config-route-map)#route-map RFD permit 20
WGxR4(config-route-map)#match ip address prefix-list RFD24
WGxR4(config-route-map)#set dampening 20 750 2000 80
WGxR4(config-route-map)#route-map RFD permit 30
WGxR4(config-route-map)#match ip address prefix-list RFD8
WGxR4(config-route-map)#set dampening 15 750 2000 60
WGxR4(config-route-map)#route-map RFD permit 40
WGxR4(config-route-map)#set dampening 10 750 2000 40
WGxR4(config-route-map)#exit
WGxR4(config)#router bgp 65003
WGxR4(config-router)#bgp dampening route-map RFD
WGxR4(config-router)#end
```

Answers to Review Questions:

Q1)   What is the purpose of route dampening?

Route dampening minimizes the impact of route flaps in downstream autonomous systems upon local and upstream autonomous systems.

Q2)   Which routes are affected by route dampening?

Route dampening affects only EBGP routes.