NLI's Cisco© CCIE Security Lab Guide

1st Edition

John Kaberna
CCIE #7146

Ray Fung

About the Author

John Kaberna is a dual Cisco Certified Internetwork Expert, CCIE #7146 (Routing/Switching and Security). John also holds the CCNP, CCDP, and CSS 1 certifications. He is the President and principal consultant for Network Consultants Group, Inc. based in San Francisco, CA. He has more than 6 years experience in Cisco networking and security including planning, designing, implementing, and troubleshooting large multiprotocol networks. John is currently designing and implementing a large retail network and is responsible for the security, performance, and management of their entire network. He also recently completed several major contracts for the U.S. Navy as a Cisco consultant and trainer. He also writes Cisco training material and is currently teaching CCIE courses for Network Learning, Inc.

Raymond Fung is a triple Cisco Certified Internetwork Expert #6832, holding designations in Routing/Switching, Security, and Communications/Services. He specializes in designing and troubleshooting service-provider and large-enterprise networks. He graduated magna cum laude with two engineering degrees from University of California, Berkeley. He is currently writing Cisco training materials for the upcoming Voice Over IP course.

Table of Contents

SECTION I LAYER 2 TECHNOLOGIES

SECTION III CISCO GENERAL SECURITY AND FIREWALLS

SECTION IV VIRTUAL PRIVATE NETWORKS

Foreword

Welcome to Network Learning Inc's CCIE Security Bootcamp. This course will help prepare you for the demanding CCIE Security lab examination. Preparing for this exam takes countless hours of preparation and a sizable investment in Cisco equipment. This course is designed to help you identify your weaknesses, train you on topics you may not be familiar with, and allow you to develop a strategy for preparing and taking the actual lab exam.

There are several "core" topics for the CCIE Security lab. You must know them extremely well and be able configure them very quickly. Not knowing these topics will almost assuredly cause you to fail the lab exam.

We consider these topics core not just because they are likely to be found on every lab, but some topics such as ATM may actually prevent you from completing another section of the lab. For example, if you need to configure an IPSec tunnel and one endpoint is on the other side of an ATM connection, it won't work without ATM and even if you configure your IPSec tunnel properly, you probably will not receive the points since your VPN is not actually functional. The same basic premise applies for routing protocols. If a routing protocol is not working properly a lot of your other sections may not work either. For example, a router is configured to use TACACS+ for authentication and the ACS server is unreachable because the network where the ACS server is connected is not in the routing table then the TACACS+ authentication would fail.

Core reachability topics:

- Frame-Relay
- Catalyst 3550 Switching
- ATM
- ISDN
- OSPF
- BGP
- EIGRP
- RIP
- Routing protocol redistribution

Core security topics:

- General Cisco Router Security
- Basic IOS Firewall Configuration
- Basic PIX Configuration
- Basic IPSec tunnels
- AAA

Many times candidates will try and "know it all" without having a firm grasp of the core technologies. There is no reason to spend hours learning Multicast if you have trouble with OSPF!

The next set of topics will typically not prevent reachability, but some of them are likely to appear on the exam. Do not ignore these topics completely in your studies, but do not devote a lot of time on them until you master the core topics.

Second tier topics:

- NTP
- SNMP
- IS-IS
- QoS

Pay close attention to any changes that Cisco may announce on their website. They may change the equipment list, software versions, or technologies tested between the time you attend this course and your lab date. The URL for the latest CCIE Security information is below.

http://www.cisco.com/warp/public/625/ccie/certifications/security.html

There may be several ambiguous questions that expect you to solve a problem. There are basically two question types. The first question type is the explicit question. It asks you to perform a specific tasks such as configure the Frame-Relay network for OSPF Area 0. This type of question is straightforward and requires little interpretation. The second question type is the problem solving question. You are given a non-specific task and usually some restrictions to accomplish the task. This type of question tests your ability to come up with a solution based on the restrictions given. Usually, this type of question involves using commands that would not normally be used in the real world. We can usually identify a handful of questions that really do not make sense in the real world since we would not be given such strange requirements. The CCIE lab attempts to test your mastery of the command set and ability to solve problems using conventional and non-conventional means.

Warning and Disclaimer

This book is designed to provide information about the Cisco CCIE Security lab examination. Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied.

The information is provided on an "as is" basis. The author, Network Consultants Group Inc., and the publisher, Network Learning Inc. shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book or from the use of the CD-ROM or programs that may accompany it.

Feedback Information

Feedback should be submitted to the following URL: www.securityie.com
That site is monitored daily by our staff. Should you have any comments, suggestions, or complaints feel free to post them on that site.

Trademark Acknowledgments

CCNA™, CCNP™, and CCIE™ are registered Trademarks of Cisco Systems, Inc

Section I

Layer 2 Technologies

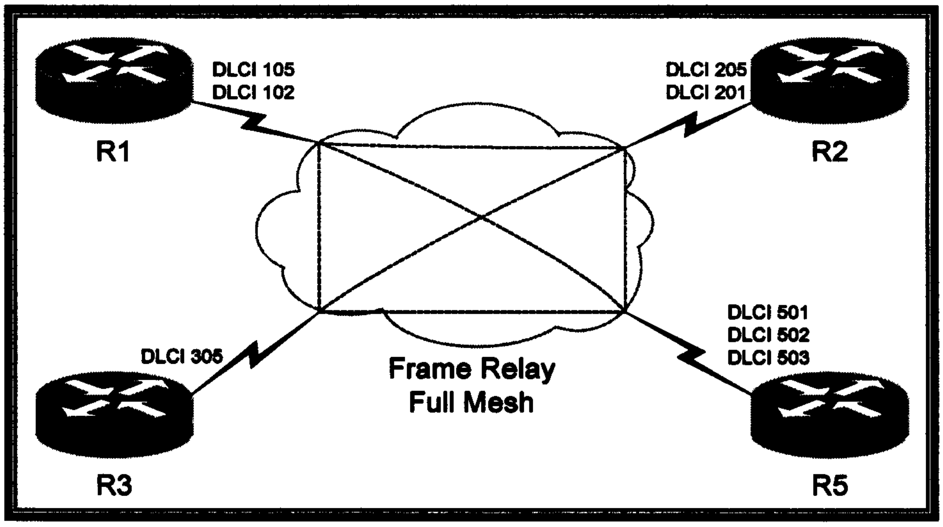
CHAPTER 1

FRAME-RELAY

INTERFACE TYPES

Prior to configuring any frame-relay network, you'll need to identify the physical interfaces to be used and their corresponding DLCI's. Figure 1.1 illustrates the topology discussed in the upcoming sections.

Figure 1.1 *Frame Relay full mesh topology*



PHYSICAL INTERFACES

Physical interfaces do not use point-to-point or point-to-multipoint subinterfaces. Physical interfaces receive all DLCI's advertised by the switch. If you create a subinterface, you need to tell that subinterface which DLCI it should use. To verify which DLCI's a physical interface receives use the `show frame pvc` command.

If your PVC is configured properly, the DLCI USAGE field should be LOCAL and PVC STATUS should be ACTIVE. Remember that both ends of the PVC need to be configured properly in order for the PVC to be active. If you have not yet configured an IP address, your DLCI USAGE should be UNUSED.

```
r5# show frame pvc

PVC Statistics for interface Serial1 (Frame Relay DTE)

DLCI = 501, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial1

  input pkts 1 output pkts 1 in bytes 30
  out bytes 30 dropped pkts 0 in FECN pkts 0
  in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
  in DE pkts 0 out DE pkts 0
  out bcast pkts 1 out bcast bytes 30
  pvc create time 00:05:51, last time pvc status changed 00:05:52

DLCI = 502, DLCI USAGE = UNUSED, PVC STATUS = ACTIVE, INTERFACE = Serial1

  input pkts 0 output pkts 4 in bytes 0
  out bytes 120 dropped pkts 0 in FECN pkts 0
  in BECN pkts 0 out FECN pkts 0 out BECN pkts 0
  in DE pkts 0 out DE pkts 0
  out bcast pkts 4 out bcast bytes 120 Num Pkts Switched 0
  pvc create time 00:08:53, last time pvc status changed 00:02:43
```

You should also check your PVC status. Active means the PVC is active and information can be exchanged. Inactive means the router's local connection to the switch is working to the frame switch, but there is a problem on the remote end. Both ends of a PVC must be up for it to be active. Deleted means the router is not receiving LMI from the frame switch or there is a layer 1 problem.

POINT-TO-POINT SUBINTERFACES

Make sure your physical interface is configured for Frame-Relay and receiving DLCI's from the switch prior to configuring a point-to-point subinterface. Point-to-point interfaces do not need to perform any DLCI to IP address mapping[1]. However, point-to-point subinterfaces require an interface DLCI to be assigned. Since all DLCI's are automatically assigned to the physical interface by default, the subinterface needs to be configured for the particular DLCI it will use.

Do not try to configure a frame map on a point-to-point subinterface! It should give you some sort of error message depending on the IOS version.

```
r2(config-subif)# frame map ip 172.16.0.1 201 broadcast
FRAME-RELAY INTERFACE-DLCI command should be used on point-to-point interfaces
```

POINT-TO-MULTIPOINT SUBINTERFACES

Like point-to-point interfaces, make sure your physical interface is configured for Frame-Relay and receiving DLCI's from the switch prior to configuring a point-to-multipoint subinterface. Point-to-multipoint subinterfaces require either an interface DLCI or a frame map. Unlike, point-to-point subinterfaces, multipoint must map DLCI's to IP addresses to work properly.

[1] Notice that only IP address mapping is mentioned and does not other protocols such as IPX. This is only because the CCIESecurity lab does not have any desktop protocols on the exam. We are only concerned with the IP protocol for the purpose of this book.

INVERSE ARP

Inverse ARP resolves the IP address of the router on the other side of the PVC. The other router's IP address does not even have to be in the same subnet as your interface! If inverse ARP maps an IP address to a DLCI it will not be removed until the router reboots or until it is manually cleared with the `clear frame-relay-inarp` command.

Inverse ARP automatically works for physical interfaces. No additional configuration is required once a PVC is configured and an IP address is assigned to the routers. In order for inverse ARP to work with a multipoint interface, an interface DLCI must be configured. Remember that point-to-point subinterfaces do not use inverse ARP or any DLCI to IP address mapping.

There are limitations with inverse ARP. Inverse ARP only works if the routers are directly connected. If a partial mesh topology is configured, the spoke routers will not be able to communicate with each other. There are three solutions to this problem.

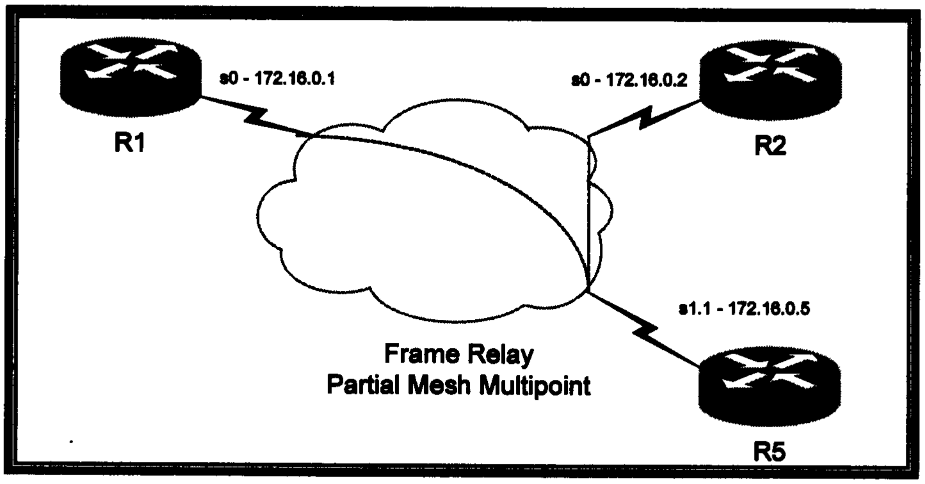
1. Configure an additional PVC between the spokes
2. Configure point-to-point subinterfaces on the hub router and create new subnets for each link
3. Configure frame maps

The first solution requires you to configure another PVC on the frame switch. It is unlikely that you will have access to the frame switch to make this change. The lab may also require that you do NOT use a full-mesh topology even if the switch is already configured for a full mesh topology.

The second solution will require you to put the spokes in different subnets and configure point-to-point connections. Since most of the IP addresses are pre-configured for you, this is probably not allowed in the lab.

The third solution is probably your only option. Once frame maps are configured, make sure you clear the dynamic ARP entries or reload the router. Shutting down the interface and then re-enabling it may also remove the dynamic entries. The proctors may not want to see both a dynamic and static mapping in your frame-relay routers.

Figure 1.2 *Frame Relay partial mesh*



Disabling inverse ARP prevents the router from requesting ARP entries on the PVC's. In Figure 1.2, R1 will send out an ARP request on its PVC's. R2 and R5 should both respond with their IP address. This may not be a desirable result. To disable a router from responding to inverse ARP requests, use the **no arp frame-relay** interface command.

Frame Maps

Frame maps are needed whenever Inverse ARP is disabled or if there is a partial mesh topology with a requirement that spoke routers be able to reach each other. To disable inverse ARP use the **no frame inverse-arp** command on the main physical interface. If inverse ARP is disabled, a frame map will need to be configured.

```
r5(config-if)# no frame inverse-arp
```

The frame map command tells the router how to reach a remote IP address and which DLCI to use. The broadcast statement allows broadcast and multicast traffic to be sent. Without the broadcast statement OSPF will not work!

```
r5(config-subif)# frame map ip 172.16.0.1 501 broadcast
r5(config-subif)# frame map ip 172.16.0.2 502 broadcast

r5# show frame map
Serial1.1 (up): ip 172.16.0.1 dlci 501(0x1F5,0x7C50), static,
              broadcast,
              CISCO, status defined, active
Serial1.1 (up): ip 172.16.0.2 dlci 502(0x1F6,0x7C60), static,
              broadcast,
              CISCO, status defined, active
```

A problem you may notice is that you cannot ping your frame-relay interface from the local router. It also needs a frame map! Just create a frame map pointing to a DLCI that is up and enter your local IP address. The example below shows R5 with a frame map to its own interface IP address.

```
r5(config-subif)# frame map ip 172.16.0.5 501
```

Never configure frame maps if an interface DLCI is configured on the same interface! Remove the interface DLCI prior to configuring a frame map and vice versa.

Frame maps also allow you to select a different frame-relay encapsulation and various compression options.

There are only two different frame-relay encapsulation types: cisco and ietf. This is normally provided by your frame-relay carrier, but in the lab you may need to use trial and error.

```
r5(config-if)# frame map ip 172.16.0.1 501 ?
  broadcast             Broadcasts should be forwarded to this address
  cisco                 Use CISCO Encapsulation
  compress              Enable TCP/IP and RTP/IP header compression
  ietf                  Use RFC1490/RFC2427 Encapsulation
  nocompress            Do not compress TCP/IP headers
  payload-compression   Use payload compression
  rtp                   RTP header compression parameters
  tcp                   TCP header compression parameters
```

RTP and TCP compression can be configured using either frame maps or the **frame-relay ip** command. The frame map option allows you to configure compression on a per PVC basis.

The active option will always compress the TCP and RTP headers. Passive only compresses the headers if the destination is also using compression. This prevents one-way compression. Beginning in version 12.1(2)T there is support for configuring the maximum number of compressed connections. The value must be between 3 and 256.

```
r5(config-if)# frame map ip 172.16.0.1 501 compress ?
  active       Always compress TCP/IP and RTP/IP headers
  connections  Maximum number of compressed TCP & RTP connections
  passive      Compress for destinations sending compressed headers
  <cr>

r5(config-if)# frame map ip 172.16.0.1 501 rtp ?
  header-compression  Enable RTP/IP compression

r5(config-if)# frame map ip 172.16.0.1 501 tcp ?
  header-compression  Enable TCP/IP header compression
```

The frame-relay ip command configures compression for the entire interface.

```
r5(config-if)# frame-relay ip rtp ?
  compression-connections  Maximum number of compressed RTP connections
  header-compression       Enable RTP/IP header compression

r5(config-if)# frame-relay ip tcp ?
  compression-connections  Maximum number of compressed TCP connections
  header-compression       Enable TCP/IP header compression
```

Cisco routers also support three different types of payload compression.

```
r5(config-if)# frame map ip 172.16.0.1 501 payload-compression ?
  FRF9              FRF9 encapsulation
  data-stream       cisco proprietary encapsulation
  packet-by-packet  cisco proprietary encapsulation
```

INTERFACE DLCI

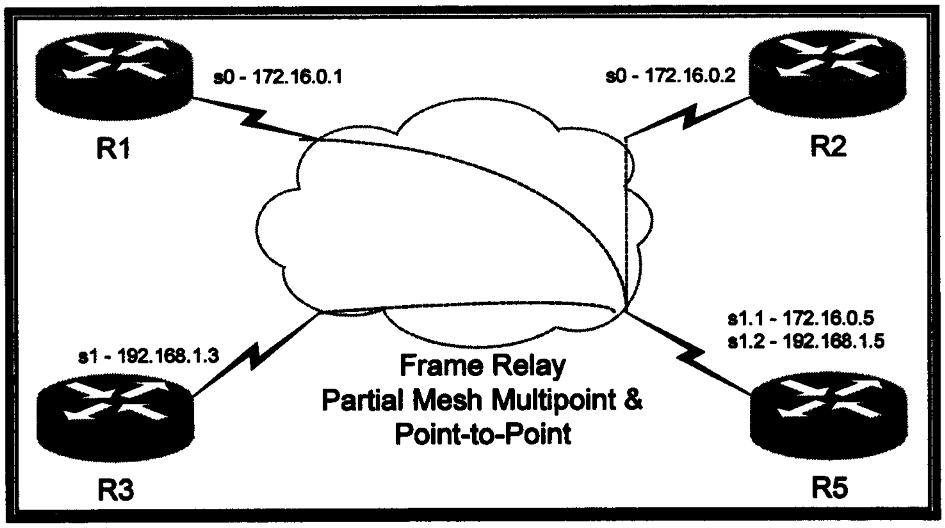
Interface DLCI's can be configured on point-to-point and point-to-multipoint subinterfaces. There is no reason to configure an interface DLCI on a physical interface since it receives all DLCI's by default. When configuring a point-to-point subinterface for frame-relay, an interface DLCI is required.

```
r5(config)# interface serial 1.2 point-to-point
r5(config-subif)# ip address 192.168.1.5 255.255.255.0
r5(config-subif)# frame interface-dlci 503

r3(config)# interface serial 1.1 point-to-point
r3(config-subif)# ip address 192.168.1.3 255.255.255.0
r3(config-subif)# frame interface-dlci 305
```

It can also be configured on a multipoint interface configured for inverse ARP. In this configuration, we let inverse ARP dynamically learn the IP address found for each DLCI.

```
r5(config)# interface serial 1.1 multipoint
r5(config-subif)# ip address 172.16.0.5 255.255.255.0
r5(config-subif)# frame-relay interface-dlci 501
r5(config-subif)# frame-relay interface-dlci 502

r5# show frame map
Serial1.1 (up): ip 172.16.0.1 dlci 501(0x1F5,0x7C50), dynamic,
              broadcast,, status defined, active
Serial1.1 (up): ip 172.16.0.2 dlci 502(0x1F6,0x7C60), dynamic,
              broadcast,, status defined, active
```

Figure 1.3 *Frame relay full mesh with point-to-point*

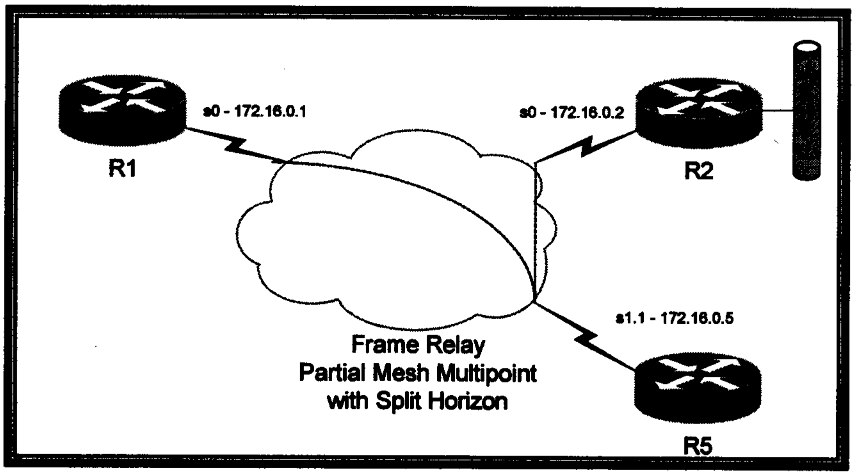


SPLIT HORIZON

Split horizon prevents a router from advertising networks out the same interface in which it was received. Distance vector routing protocols such as RIP and IGRP as well as hybrid protocols such as EIGRP will not work properly in a frame-relay multipoint network if split horizon is enabled. OSPF does not use split horizon. Split horizon is automatically enabled for point-to-point and multipoint interfaces. It is disabled for physical interfaces. Disabling split horizon for IP protocols requires the **`no ip split-horizon interface`** command. For EIGRP, specify the EIGRP AS number as shown in the second example.

```
r5(config)# interface serial 1.1
r5(config-subif)# no ip split-horizon

r5(config-subif)# no ip split-horizon eigrp 1
```

Figure 1.4 *Frame Relay Partial Mesh Multipoint with Split Horizon*



In Figure 1.4, R2 is advertising network 10.1.1.0 to R5. With split horizon enabled, R5 cannot send this route to R1 because it cannot send routing updates out the same interface it was received. Figure 1.4. Frame relay multipoint with Split Horizon

Before we disable split horizon, observe the routing tables on R2, R5, and R1. R2 has an ethernet configured with a 10.1.1.0 network. This 10.0.0.0 route appears in R5's routing table. Since RIP is classful it advertises the route on the classful boundary.

```
r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial1.1
R    10.0.0.0/8 [120/1] via 172.16.0.2, 00:00:09, Serial1.1
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
```

The 10.0.0.0 /8 route does not show up on R1. Check the routing table after disabling split horizon on the multipoint interface on R5.

```
r5(config-subif)# no ip split-horizon
```

The route now shows up on R1.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
R    10.0.0.0/8 [120/2] via 172.16.0.5, 00:00:00, Serial0
```

Although the routing looks correct, there may still be a problem. With split horizon disabled, the network is now susceptible to routing loops. To fix this problem, use distribute-lists to control the traffic received from a particular router. Notice that before we configure any distribute lists, R5 is advertising the 10.0.0.0 network back to R2! This may cause have a temporary routing loop.

```
r2# debug ip rip
RIP protocol debugging is on

03:58:00: RIP: received v1 update from 172.16.0.5 on Serial0
03:58:00:      172.16.0.0 in 1 hops
03:58:00:      10.0.0.0 in 2 hops
03:58:02: RIP: sending request on Ethernet0 to 255.255.255.255
03:58:02: RIP: sending request on Serial0 to 255.255.255.255
03:58:02: RIP: sending v1 update to 255.255.255.255 via Ethernet0 (10.1.1.1)
03:58:02: RIP: build update entries
03:58:02:       network 172.16.0.0 metric 1
03:58:02: RIP: sending v1 update to 255.255.255.255 via Serial0 (172.16.0.2)
03:58:02: RIP: build update entries
03:58:02:       network 10.0.0.0 metric 1
03:58:02:       subnet 172.16.0.0 metric 1
```

In order to prevent routing loops we configure an access list and a distribute list to prevent R5 from advertising R2's own network back to R2.

```
r5(config)# access-list 1 deny 10.0.0.0 0.255.255.255
r5(config)# access-list 1 permit any
r5(config-router)# distribute-list 1 out Serial1.1
r2# debug ip rip
04:00:44: RIP: received v1 update from 172.16.0.5 on Serial0
04:00:44:      172.16.0.0 in 1 hops
04:04:44: RIP: sending v1 update to 255.255.255.255 via Serial0 (172.16.0.2)
04:04:44: RIP: build update entries
```

```
04:04:44:      network 10.0.0.0 metric 1
04:04:44:      subnet 172.16.0.0 metric 1
```

LMI Types

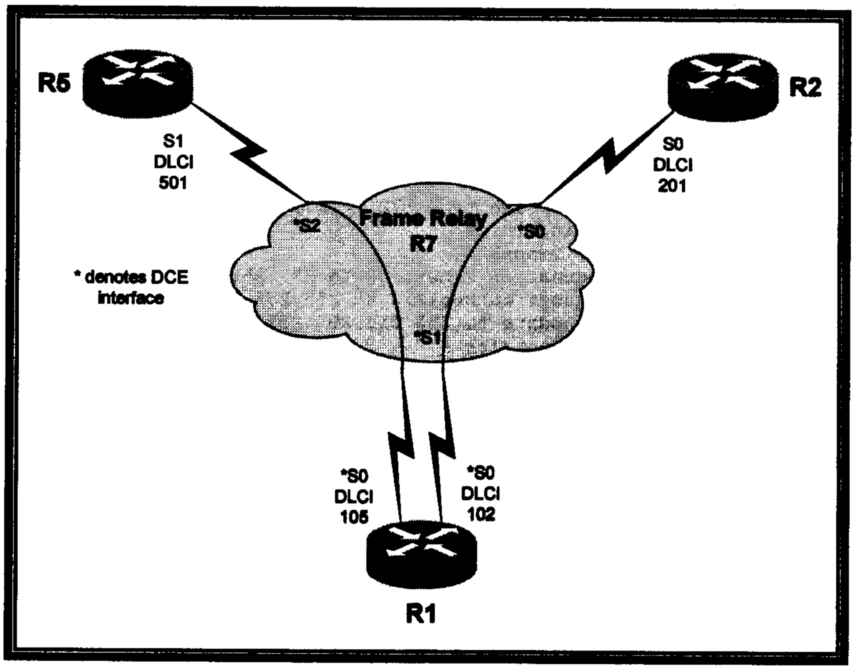
There are 3 different LMI types. Although these are controlled by the frame switch, the exam may provide you clues as to which type you need to use. Worse case scenario, you can use trial-and-error to find out which LMI types the frame-switch is using. CISCO is the default LMI type. It sends its LMI status on DLCI 0. ANSI is the most common LMI type in provider networks. It is also called Annex D. ANSI uses DLCI 1023 for LMI status. Q933a is also called Annex A and uses DLCI 0 for LMI status. If the exam instructed you to use Annex A, Annex D, or an LMI type that uses DLCI 1023 for LMI status, you should know which type to configure.

Frame Switching

In order to configure frame-relay routers, you must have a frame switch. In the real world, this is typically provided by the circuit provider such as AT&T or Sprint. In the CCIE lab, a router is configured to provide this service. Configuring a router as a frame-switch is not part of the CCIE lab, but if you are going to build a home lab you will need to know how to configure frame-switching.

In Figure 1.5, we configured R7 as a frame switch and used the DLCI's shown in the drawing.

Figure 1.5 *Frame Switching*



Step 1 Configure the interfaces for frame-relay encapsulation. It is also recommended that you configure descriptions so your configurations make sense.

```
r7(config)# interface Serial0
r7(config-if)# encapsulation frame-relay
r7(config-if)# description Router 2 serial 0
```

Step 2 Configure the frame-relay interface type for DCE. This is the frame-relay type which has nothing to do with the actual DCE end of the cable. Your frame switch is the only router that should have this command.

```
r7(config-if)# frame-relay intf-type dce
```

Step 3 Configure your frame routes. This is how you map your two DLCI's to form a PVC.

```
r7(config-if)# frame-relay route 201 interface Serial1 102
```

Step 4 Configure a clock rate for any interfaces that are layer 1 DCE (not frame-relay which is layer 2). I recommend configuring a clock rate on every interface. If you configure a clock rate on a DTE you will get a harmless error that you can ignore.

```
r7(config-if)# clockrate 56000
```

Step 5 Repeat steps 1-4 for each router. Your final configuration should look like the one below.

```
interface Serial0
 description Router 2 serial 0
 encapsulation frame-relay
 frame-relay intf-type dce
 frame-relay route 201 interface Serial1 102
!
interface Serial1
 description Router 1 serial 0
 encapsulation frame-relay
 frame-relay intf-type dce
 frame-relay route 102 interface Serial0 201
 frame-relay route 105 interface Serial2 501
!
interface Serial2
 description Router 5 serial 1
 encapsulation frame-relay
 clockrate 56000
 frame-relay intf-type dce
 frame-relay route 501 interface Serial1 105
```

Typical Gotchas!

- Missing a clock rate on the DCE end of a serial interface
- Frame maps configured on the spokes for the hub and not the other spoke(s)
- Missing broadcast at the end of a frame map
- Frame switch is misconfigured
- Subnet masks are not the same on all routers in a multipoint network
- Split horizon is enabled on a multipoint network running EIGRP

CHAPTER 2

ATM

ATM technology is quite complex and difficult. This section will explain what you need to know to get an ATM configuration working. For ATM theory, we recommend the book Cisco ATM Solutions from Cisco Press. It has an excellent explanation of ATM theory as well as how to configure ATM on Cisco routers and ATM switches.

TERMS

In order to understand how to configure ATM there are a few terms you need to understand. ATM is based on either PVC's or SVC's. A PVC is a permanent virtual circuit. As its name implies, the circuit is up all the time. An SVC is a switched virtual circuit. SVC's are brought up on an as needed basis in a similar way to ISDN or a regular phone call.

RFC 2684 VERSUS RFC 2225

RFC 2684 is a way to transport various upper layer protocols (IP, IPX, etc.) over an ATM network. It requires manual configuration to map the layer 3 addresses to ATM. This is very similar to frame-relay. RFC 2225 is inverse ARP for ATM. It only works with the IP and IPX protocol[2], so if you need to transport other protocols you'll need to configure an RFC 2684 connection. Both RFC types can be configured for either PVC's or SVC's.

VCD, VPI, AND VCI

PVC's consist of a VCD, VPI, and VCI. The VCD is a virtual circuit descriptor. The VCD uniquely identifies the PVC in the router. The VCD must be unique on the router. It is similar to a DLCI in frame-relay. The VPI is a virtual path identifier. It is analogous to a highway with several lanes. The VCI is a virtual channel identifier. It is analogous to a single lane on a highway. The VPI and VCI is a pair and must be unique for the ATM interface on the router. The same VPI/VCI pair can be used on the same router, but only on a different interface. The ATM provider (or the proctor in the case of the CCIE lab) assigns the VPI/VCI pair. The ATM switch will also advertise the VPI/VCI pair if ILMI is configured.

[2] Inverse ARP for IPX requires that you configure PVC's the "new" way. Check your IOS version to see if it supports inverse ARP for IPX. IPX

ILMI AND QSAAL

ILMI and QSAAL are two well-known PVC's that may need to be configured depending on the specific requirements for the connection. ILMI is very useful for troubleshooting and providing information about the ATM connection. It uses the VPI/VCI pair 0\16. It can also provide other useful information such the NSAP prefix, IP address of the ATM switch, interface on the ATM switch, etc.

When configuring ATM, the first step should be to enable ILMI to verify you are communicating with the ATM switch. Consider configuring PVC autodiscovery to verify the PVC's documented in your lab exam are in fact what is configured on the switch.

Once ILMI is configured, you can verify that you are communicating with the ATM switch. The command show atm ilmi-status displays your interface, the VPI/VCI pair, the ATM switch IP address, the ATM switch interface, and the ILMI state.

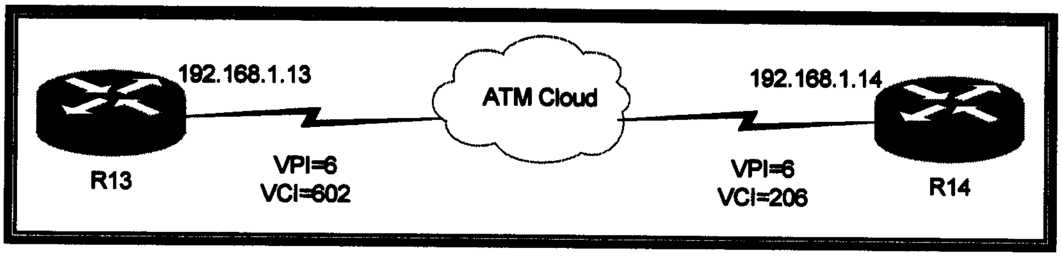
```
r13# show atm ilmi-status

Interface : ATM2/0 Interface Type : Private UNI (User-side)
ILMI VCC : (0, 16) ILMI Keepalive : Disabled
ILMI State:          UpAndNormal
Peer IP Addr:        10.10.1.100      Peer IF Name:      ATM1/1/2
Peer MaxVPIbits:     8                Peer MaxVCIbits:   14
Active Prefix(s) :
47.0091.8100.0000.00e0.1e79.7201
```

The second well-known PVC is QSAAL. It uses the VPI/VCI pair 0\5. This PVC should only be configured in an SVC network or when your PVC needs QoS functionality. We advise you not to configure this PVC unless it is necessary. Configuring unnecessary commands may not demonstrate the mastery that the proctors are probably looking for. If they see additional commands that are not necessary they may choose to not give you the points for that section. You need to understand every command you are entering and why it is there. They do not want to see that you can blindly regurgitate sample configurations from Cisco's website. However, we should mention that we feel ILMI is not a requirement for most ATM configurations, it is always somewhat useful and the proctors should not be concerned if you have it configured.

```
r13(config)# interface ATM2/0
r13(config-if)# atm pvc 2 0 5 qsaal
```

PVC CONFIGURATION

Figure 2.1 *Basic ATM layout*



SUBINTERFACES

If you configure ATM point-to-point subinterfaces, you do not need to configure any layer 2 to layer 3 mapping. Just like Frame-Relay, there is no need to configure static maps with a point-to-point subinterface. If your IP address is already configured for you on the physical interface do not create a new subinterface. Never change interface types if the routers are pre-configured with an IP address. This option is only available to you if you have the option to configure ATM from scratch or if the router was already configured with a point-to-point subinterface.

```
r14(config)# interface ATM2/0.1 point-to-point
r14(config-subif)# ip address 192.168.1.14 255.255.255.0
r14(config-subif)# pvc 6/206
r14(config-if-atm-vc)#

r13(config)# interface ATM2/0.1 point-to-point
r13(config-subif)# ip address 192.168.1.13 255.255.255.0
r13(config-subif)# pvc 6/602
r13(config-if-atm-vc)#

r13# ping 192.168.1.14

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.14, timeout is 2 seconds:
!!!!!
```

AUTODISCOVERY

The first step to autodiscovery is to create the well-known VPI/VCI for ILMI. Then configure the autodiscovery.

```
r13(config)# interface ATM2/0
r13(config-if)# atm pvc 1 0 16 ilmi
r13(config-if)# atm ilmi-pvc-discovery
```

Now check to see if your router discovered the PVC. As expected, it shows up with the VPI/VCI of 6\ 602. You can tell that the PVC is learned via autodiscovery by the "type" field. The type field is PVC-D. The "D" obviously stands for discovery. Autodiscovery will randomly assign the VCD number.

```
r13# show atm pvc
           VCD /                                           Peak  Avg/Min Burst
Interface  Name        VPI   VCI  Type    Encaps    SC    Kbps    Kbps    Cells
   Sts
2/0         1           0    16   PVC     ILMI      UBR   155000
   UP
2/0         5           6   602   PVC-D   SNAP      UBR   155000
   UP
```

We did notice some interesting behavior when trying to configure autodiscovery and the "new" way for PVC's. The configurations were entered without any errors. However, when we checked the configuration they were gone! When we used **map-groups** and **map-lists** the configurations were there and worked just fine. This may be version dependent, so your results may vary.

MAP-GROUP AND MAP-LISTS

Using map-groups and map-lists are similar to using frame-maps in frame-relay. We need to map the router's IP address on the opposite end of our PVC. The VCD is similar to a frame-relay DLCI. However, the VPI/VCI pair has no equivalent in frame-relay. However, before you can create a map-list or map-group, you must configure your PVC(s). Your PVC can either be autodiscovered or it will be given to your by the lab instructions. The first number in the **atm pvc** command is the VC or VCD. The second number is the VPI and third number is the VCI. The encapsulation type is "aal5snap" and is probably the only one you will need to use on the lab. It allows all protocols to use the same circuit as opposed to "aal5mux" which requires a different circuit for each layer 3 protocol. The other encapsulation types are ilmi, qsaal, aal34smds, and NLPID.

```
r13(config)# interface ATM2/0
r13(config-if)# atm pvc 1 6 602 aal5snap

r14(config)# interface ATM2/0
r14(config-if)# atm pvc 1 6 206 aal5snap
```

First, configure a map-list with an arbitrary name. However, you should choose one that helps you identify the connection especially if you are going to have several map-lists configured. In our example, we chose RFC2684 because we are configuring a multiprotocol encapsulation connection. Plus, we thought this would impress the proctor that we know our RFC's! Prior to configuring the map-list, you should identify the VCD used by the PVC. Use the **show atm pvc** command as shown in the previous section.

```
r13(config)# map-list RFC2684
r13(config-map-list)# ip 192.168.1.14 atm-vc 5 broadcast

r13# show atm map
Map list RFC2684 : PERMANENT
ip 192.168.1.14 maps to VC 5
        , broadcast

r14(config)# map-list RFC2684
r14(config-map-list)# ip 192.168.1.13 atm-vc 6 broadcast

r14# show atm map
Map list RFC2684 : PERMANENT
ip 192.168.1.13 maps to VC 6
        , broadcast

Configuring the map-list is not enough.  It must be applied to the ATM
    interface.  This is similar to configuring an access-list and applying it
    to an interface with an access-group command.

r13(config)# interface ATM2/0
r13(config-if)# map-group RFC2684

r14(config)# interface ATM2/0
r14(config-if)# map-group RFC2684
```

"NEW" WAY

PVC's can be configured without using map-group or map-lists. The **pvc** command, which first became available in version 11.3, will eventually replace the **atm pvc** command. For now, the **atm pvc** command is still available and supported. Do not use the "new" way on one end of a PVC and the old map-list way on the other end. You should notice that we did not configure a VCD. The router automatically assigned VCD number "2" to the connection.

```
r13(config)# interface ATM2/0
r13(config-if)# pvc 6/602
r13(config-if-atm-vc)# protocol ip 192.168.1.14 broadcast

r13# show atm pvc
            VCD /                                         Peak  Avg/Min Burst
```

```
Interface   Name        VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells
    Sts
2/0        1             0     16  PVC    ILMI     UBR  155000
    UP
2/0        2             6    602  PVC    SNAP     UBR  155000
    UP

r14(config)# interface ATM2/0
r14(config-if)# pvc 6/206
r14(config-if-atm-vc)# protocol ip 192.168.1.13 broadcast

r13# show atm pvc
            VCD /                                       Peak  Avg/Min Burst
Interface   Name        VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells
    Sts
2/0        1             0     16  PVC    ILMI     UBR  155000
    UP
2/0        2             6    206  PVC    SNAP     UBR  155000
    UP
```

Instead of letting the router assign a VCD, you can configure a more descriptive name for the connection. You should notice that our VCD/Name is now "toR14."

```
r13(config)# interface ATM2/0
r13(config-if)# pvc toR14 6/602
r13(config-if-atm-vc)# protocol ip 192.168.1.14 broadcast
r13# show atm pvc
            VCD /                                       Peak  Avg/Min Burst
Interface   Name        VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells
    Sts
2/0        1             0     16  PVC    ILMI     UBR  155000
    UP
2/0        toR14         6    602  PVC    SNAP     UBR  155000
    UP
```

INVERSE ARP

Inverse ARP is also called Classical IP as described in RFC 2225. Classical IP can be either PVC or SVC based. Inverse ARP can be configured three different ways. Classical IP over PVC's can be created using **map-groups** and **map-lists** or with the **pvc** command. Classical IP over SVC's requires an ARP server and ARP client(s).

CLASSICAL IP OVER PVC'S (OLD WAY)

Configuring Classical IP over PVC's still requires at least one PVC be created. The mapping of the IP address to the ATM VC is the part that is done dynamically with Classical IP.

```
r13(config)# interface ATM2/0
r13(config-if)# atm pvc 2 6 602 aal5snap inarp

r14(config)# interface ATM2/0
r14(config-if)# atm pvc 2 6 206 aal5snap inarp
```

One of the interesting things about Classical IP is that if you enter the **show atm map** command there is no mention of the mapping. That is because this command only shows static mappings. The only command we could find that will show us if ARP is working is the regular **show arp** command! This is after you ping or send some traffic to cause the router to ARP for the address.

```
r14# show arp
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
```

```
Internet  192.168.1.13           1    6 / 206        ATM    ATM2/0
```

Classical IP over PVC's (New Way)

Configuring Classical IP using the new pvc command actually requires three commands instead of two. We also used the optional description field in our **pvc** command. Note that the VC does not go away, it's just dynamically created. The description field actually replaces the VC number. The only way you can tell what VC is being used is by entering the **show atm map** command.

```
r13(config)# interface ATM2/0
r13(config-if)# pvc toR14 6/602
r13(config-if-atm-vc)# protocol ip inarp broadcast

r14(config)# interface ATM2/0
r14(config-if)# pvc toR13 6/206
r14(config-if-atm-vc)# protocol ip inarp broadcast
```

Unlike the old way of configuring PVC's, the **show atm map** command actually shows the dynamic mapping! It certainly does not make much sense since the IOS help tells us that **show atm map** is only for static mappings. The **show arp** command does not give us anything this time. As we've seen the behavior does not seem to make much sense or have any consistency. Just know how to configure both ways in case you are asked to choose one way over another.

```
r14# show atm ?
  arp-server          ATM ARP Server Table
  bundle              ATM bundle information
  class-links         ATM vc-class links
  ilmi-configuration  Display Top level ILMI
  ilmi-status         Display ATM Interface ILMI information
  interface           Interfaces and ATM information
  map                 ATM static mapping
  pvc                 ATM PVC information
  signalling          ATM Signalling commands
  svc                 ATM SVC information
  traffic             ATM statistics
  vc                  ATM VC information
  vp                  ATM VP information

r14# show atm map
Map list ATM2/0_ATM_INARP : DYNAMIC
ip 192.168.1.13 maps to VC 4, VPI 6, VCI 206, ATM2/0

r14# show arp
```

SVC Configuration

There are several ways to configure SVC's. We will only cover the most common: Classical IP over SVC's.

Classical IP over SVC's

Classical IP over SVC's requires an ARP server and one or more ARP clients. The server handles mapping IP addresses to ATM VC's for the network.

Prior to configuring any SVC's you must configure the well-known PVC for QSAAL. Even if you are using subinterfaces, this command must be entered on the main physical interface.

```
r13(config)# interface ATM2/0
r13(config-if)# atm pvc 2 0 5 qsaal
```

SVC's use a 20-byte NSAP address. Each ATM device in the ATM cloud must have a unique NSAP address. Since these can be manually configured, we use the router number for the purposes of a lab. In our example, we are making R13 the ARP Server and R14 is the client. The NSAP address is something we created using the router number and enough 0's to make it 20 bytes. If you have as much trouble as we do counting the 20-bytes, we suggest finding an example on the Documentation CD and copying their zeros. The rest of the configuration should be self-explanatory.

```
r13(config)# interface ATM2/0
r13(config-if)# atm nsap 13.131300000000000000000000.000000000000.00
r13(config-if)# atm arp-server self

r14(config)# interface ATM2/0
r14(config-if)# atm nsap 14.141400000000000000000000.000000000000.00
r14(config-if)# atm arp-server 13.131300000000000000000000.000000000000.00
```

TYPICAL GOTCHAS!

- Using the old pvc command on one end of the PVC and the new way on the other end
- Missing map-group on the ATM interface
- Map-list does not have broadcast
- Map-list has the wrong IP address
- Map-list references the wrong VC
- Split horizon is enabled on a multipoint network
- Trying to enable a static mapping when a dynamic mapping already exists

CHAPTER 3

ISDN

ISDN is a topic that typically contributes heavily to a candidate's failure. The technology is not that difficult if it is approached in a modular manner. Frequently I notice students working on an advanced topic such as Dialer Callback when their SPID's are not even valid! If you configure ISDN one step at a time and check your work as you go, you'll know exactly which part of the configuration is not working.

The use of ISDN to connect routers together is frequently called Dial on-demand routing or DDR. DDR can also use analog circuits. You should also notice that the terms ISDN and BRI seem to be used interchangeably. ISDN comes in two flavors: BRI and PRI. A BRI (basic rate interface) consists of two Bearer channels capable at operating up to 64kbps. There is also one 16kbps D channel. The D channel handles tasks such as call setup and other circuit controls. The B (bearer) channels are simply used to transmit voice or data. A PRI consists of 23 B channels and one 64kpbs D channel. We typically refer to BRI when we are specifically talking about the interface itself. We use the term ISDN or DDR to discuss basic dialup concepts and configuration.

Minimum Configuration

There are six commands that must be configured in order for an ISDN router to dial and connect to another ISDN router.

1. ISDN switch type
2. ISDN SPID's (not required in some locations) and dial string
3. Dialer-list
4. Dialer-group
5. IP address for the interface
6. Dial string

ISDN Switch Type

Configure your ISDN switch type. Your ISDN circuit provider will have to give this to you. In the case of the CCIE lab, this will be your proctor. You can configure this command globally or on the interface.

```
r5(config)# isdn switch-type basic-net3

r6(config)# isdn switch-type basic-net3
```

SERVICE PROFILE IDENTIFIER (SPID)

Your ISDN provider will also have to provide SPID's (some locations particularly outside the U.S. do not use SPID's) and dial numbers. Once you configure your SPID's, make sure they are valid! This is probably the most common mistake when configuring ISDN. The number after the SPID is the LDN (local dial number), which is the same number as the dial string. It is not required, but we recommend using it because of its effect on multilink.

```
r5(config)# interface bri0
r5(config-if)# isdn spid1 24899133330101 9913333
r5(config-if)# isdn spid2 24899144440101 9914444

r6(config)# interface bri0
r6(config-if)# isdn spid1 24899111110101 9911111
r6(config-if)# isdn spid2 24899122220101 9912222
```

When using the `show isdn status` command there are three things you should check.

1. Verify the SPID's are valid.
2. Verify layer 1 status is ACTIVE, which indicates your connection is cabled properly and your interface is enabled.
3. Verify you have MULTIPLE_FRAME_ESTABLISHED, which indicates your layer 2 is functioning properly between your router and the ISDN switch. If you do not have these three things, DO NOT continue with your ISDN configuration!

```
r5# show isdn status
Global ISDN Switchtype = basic-ni
ISDN BRI0 interface
        dsl 0, interface ISDN Switchtype = basic-ni
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 64, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
        TEI = 65, Ces = 2, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Spid Status:
        TEI 64, ces = 1, state = 8(established)
            spid1 configured, spid1 sent, spid1 valid
            Endpoint ID Info: epsf = 0, usid = 70, tid = 1
        TEI 65, ces = 2, state = 5(init)
            spid2 configured, spid2 sent, spid2 valid
            Endpoint ID Info: epsf = 0, usid = 70, tid = 2
    Layer 3 Status:
        0 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 1
        CCB:callid=0x923E, sapi=0x0, ces=0x1, B-chan=1
    The Free Channel Mask:  0x80000002
    Total Allocated ISDN CCBs = 1
```

DIALER-LIST

Now that our SPID's are valid and we are communicating with the ISDN switch, let's move on to tell the router when to dial. An ISDN router dials when it sees "interesting traffic." Interesting traffic is defined by the administrator configuring the router. It can be as broad as any IP traffic or as narrow as SMTP traffic from a specific source to a specific destination. Pay attention to the lab requirements to properly determine what should be considered interesting traffic.

The first example is a basic configuration to permit all IP traffic. The number "1" after dialer-list indicates the dialer-group that is belongs to. Dialer-groups are explained in the next section. After the protocol we have the choice of permit, deny, or list.

```
r5(config)# dialer-list 1 protocol ip permit
```

This example is the same as the first except we are being more granular with what we consider interesting traffic. We only want Telnet traffic from the 10.0.0.0 network to the 192.168.1.0 network to bring up the ISDN line on R5. So, we first must configure an access-list. Then, we use the list option after the protocol to tell the dialer that it can only dial if there is a match on this access-list. Also, you are only permitted a single dialer-list command per protocol. If we entered the **dialer-list 1 protocol ip permit** command as shown in the previous example and then the command shown below, the new dialer-list would overwrite the first command.

```
r5(config)# access-list 100 permit tcp 10.0.0.0 0.255.255.255 192.168.1.0
    0.0.0.255 eq 23
r5(config)# dialer-list 1 protocol ip list 100
```

DIALER-GROUP

Our interesting traffic is defined, but it must be applied to the interface. Remember that you can have multiple dialer-lists and apply different lists to different ISDN interfaces. Use the same number for your dialer-group as your dialer-list. But, you can only apply one dialer-group to an interface. If you accidentally enter another dialer-group number, the router will use the last one entered.

```
r5(config)# interface bri0
r5(config-if)# dialer-group 1

r6(config)# interface bri0
r6(config-if)# dialer-group 1
```

Dialer-lists and dialer-groups are only required on one side of the ISDN link. If one router is not configured for interesting traffic it will simply never dial. However, it will still answer and connect an incoming call. This is useful for hub and spoke environments where only the spokes need to initiate calls.

IP ADDRESS

Configure an IP address on each router. This may already be done for you in the real lab. If not, make sure you use the same subnet masks. This may seem obvious, but you will find that careless mistakes come very easy under pressure.

```
r5(config)# interface bri0
r5(config-if)# ip address 192.168.1.5 255.255.255.252

r6(config)# interface bri0
r6(config-if)# ip address 192.168.1.6 255.255.255.252
```

DIAL STRING

This is simply the "phone number" of the other router. This only needs to be configured on the calling router. If a router does not need to place calls, it does not need a dial string. In our example, we want either R5 or R6 to be able to place a call.

```
r5(config)# interface bri0
r5(config-if)# dial string 4930622

r6(config)# interface bri0
r6(config-if)# dial string 4930624
```

ADVANCED CONFIGURATIONS

DIALER-MAP

The dialer map tells the router which number to dial when it needs to reach the configured IP address. Like dialer strings, dialer maps only need to be configured on routers that you want to initiate a call. In our example, we want both R5 and R6 to be able to place a call.

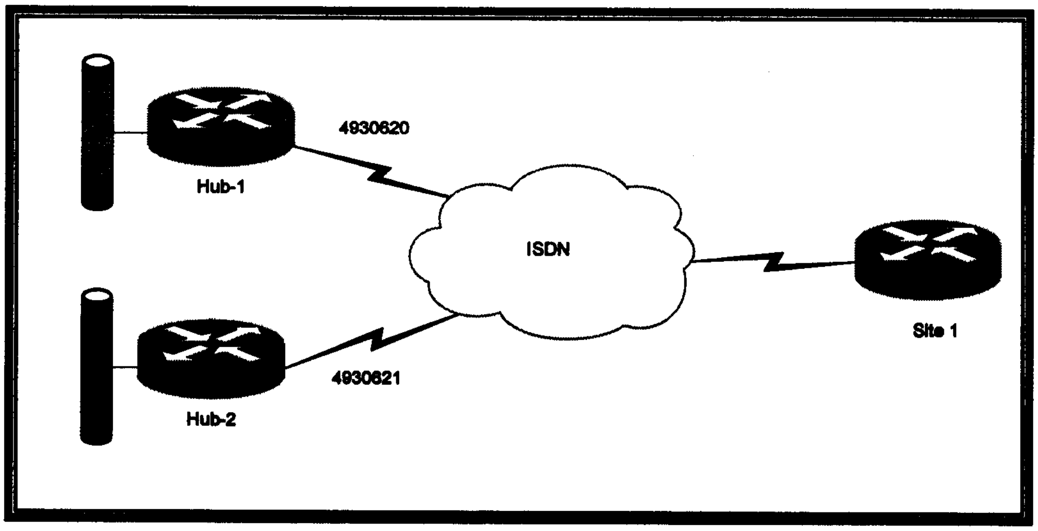
```
r5(config)# interface bri0
r5(config-if)# dialer map ip 192.168.1.6 4930622

r6(config)# interface bri0
r6(config-if)# dialer map ip 192.168.1.5 4930624
```

Dialer maps are typically not configured with such simplicity. The above configuration does not provide more functionality than a simple dial string. Dialer maps address the limitations of the dial string command. The limitations of the dial string command include the following:

- No support for multiple next hop routers
- No support for PPP multilink
- No support for ISDN callback

A router will attempt to dial all the numbers configured with the dial string command until one answers. As soon as one answers the router stops dialing. This will not work if the BRI interface needs to be able to call a specific router to reach a specific network. As shown in figure 3.1, what will happen if only a dial string was configured at Site-1? How would Site-1 know that to reach 10.1.0.0 that it needs to dial 4930620? There is no way for it to know which number to dial without dialer maps (or using dialer profiles as explained later in this chapter).

Figure 3.1 *ISDN with multiple next hop routers*



Idle Timeout

Once a link is idle (no interesting traffic) for a period of time, the BRI circuit will eventually disconnect. This saves connection costs. The default is 120 seconds and can be changed to any value from 1 second to 2147483 seconds. In our example, we want the ISDN link to disconnect after 60 seconds instead of the default of 120 seconds.

```
r5(config)# interface bri0
r5(config-if)# dialer idle-timeout 60
```

Fast Idle

If the BRI interface needs to be used to place another call, this command effectively can terminate a call without waiting for the idle timeout. This prevents the router from having to wait a lengthy idle timeout to place a call. The default is 20 seconds and can be changed to any value from 1 second to 2147483 seconds. In our example, we want the BRI interface to be available after 30 seconds of idle time to be used for another call.

```
r5(config)# interface bri0
r5(config-if)# dialer fast-idle 30
```

HDLC

The default encapsulation for a BRI and Serial interface is HDLC. The version of HDLC used by Cisco routers is proprietary and not industry standard. Since HDLC does not support authentication, PPP is typically a more secure choice. Since HDLC is the default encapsulation, you do not need to configure it. However, if you configured PPP and then wanted to change back to HDLC use the `encapsulation hdlc` interface command.

HDLC supports STAC compression using the **compress stac** interface command. This is the only supported compression type for HDLC. Compression must be configured on both sides of a link. HDLC.

PPP

The default encapsulation for a BRI interface is HDLC. PPP is typically recommended for dialup related services. PPP is also a standard protocol so it is recommended when using ISDN or serial connections in a mixed-vendor environment. Most of these configurations are also valid when running PPP over a serial interface. If you know how to configure PPP for ISDN you should be able to configure it for serial interfaces as well.

```
r5(config)# interface bri0
r5(config-if)# encapsulation ppp
```

PPP has several options that can be configured. The most important are authentication, link quality, callback, and multilink.

```
r5(config-if)# ppp ?
  authentication  Set PPP link authentication method
  bap             Set BAP bandwidth allocation parameters
  bridge          Enable PPP bridge translation
  callback        Set PPP link callback option
  chap            Set CHAP authentication parameters
  encrypt         Enable PPP encryption
  ipcp            Set IPCP negotiation options
```

```
lcp              PPP LCP configuration
link             Set miscellaneous link parameters
max-bad-auth     Allow multiple authentication failures
multilink        Make interface multilink capable
pap              Set PAP authentication parameters
quality          Set minimum Link Quality before link is down
reliable-link    Use LAPB with PPP to provide a reliable link
timeout          Set PPP timeout parameters
use-tacacs       Use TACACS to verify PPP authentications
```

CHAP Authentication

The other major benefit of using PPP is authentication. An ISDN connection can be authenticated with either CHAP or PAP.

CHAP encrypts its authentication packets. By default, a router configured for CHAP, sends its hostname as its username. This is can be changed however. The interesting thing about CHAP is that the authenticating router must have a local user account created for the calling router. By default, the routers both authenticate each other regardless of which router initiates the call.

```
r5(config)# username r6 password cisco
r5(config)# interface bri0
r5(config-if)# ppp authentication chap

r6(config)# username r5 password cisco
r6(config)# interface bri0
r6(config-if)# ppp authentication chap
```

If you are using dialer maps, you will need to use the "name" option. If you enter the wrong name or leave it blank, you may receive errors in your debugs that say "no matching dialer map" although it may still dial and appear to work properly.

```
r5(config-if)# dialer map ip 192.168.1.6 name r6 4930622

r6(config-if)# dialer map ip 192.168.1.5 name r5 4930624
```

You can control both the router's username and its password. Remember that by default, it sends its hostname and the password is what is in its own local database. So, you configure a username and password for the router you are authenticating. How does the router know what password to send? It sends the same password that it accepts! For example, R5 has a local account for R6 with a password of "cisco." When R5 authenticates with R6 it sends the password from its local database, which is obviously "cisco." Although it does not matter which password you choose, they must be the same on both sides UNLESS you specifically configure the password to send as shown in the example below. Make sure you create a new account on R6 so it can successfully authenticate R5 with the newly created username and password.

```
r5(config-if)# ppp chap hostname router5
r5(config-if)# ppp chap password ccie

r6(config)# username router5 password ccie
```

PAP Authentication

PAP sends its authentication packets via clear text and is therefore much less secure than CHAP. Unlike CHAP, PAP requires a username and password to be explicitly defined. It does not default to using the router's hostname as its username. Both routers still require a local account to be created just like CHAP.

```
r5(config-if)# ppp pap sent-username r5 password 0 cisco

r6(config-if)# ppp pap sent-username r6 password 0 cisco
```

You should have noticed the "0" immediately after "password." This simply means that the password you are about to enter ("cisco" in our example) is not yet encrypted. If you were entering an encrypted password you would use the number "7" instead of "0."

Link Quality Reliable Link

PPP reliable link allows the Link Access Procedure, Balanced (LAPB) to use numbered mode negotiation. This allows for the retransmission of packets across the link if the router detects that packets have errors. There are some limitations to reliable link. You cannot use reliable link in conjunction with multilink and it also results in additional overhead. Cisco recommends the use of compression when using reliable link to offset this additional overhead. The other end of a link must also be configured for reliable link. The router will not try and send packets reliably in one direction. The command only enables the router to attempt to negotiate the use of a reliable link.

```
r5(config-if)# ppp reliable-link

r6(config-if)# ppp reliable-link
```

Link Quality Monitoring

Link Quality Monitoring (LQM) measures the link quality. If a PPP link configured for LQM does not conform to the configured percentage, the router will shut down the link. The percentages are calculated for both incoming and outgoing traffic. For example, if the router is configured with a percentage of 90, as soon as the router determines that more than 10% of the packets and bytes contain errors, the router will shut down the PPP link. LQM uses Link Quality Reports (LQRs) instead of keepalives. LQR's are sent at the same interval as keepalives.

```
r5(config-if)# ppp quality ?
  <0-100>  Percent of traffic successful
```

Compression

Cisco routers running PPP support multiple compression types. STAC compression is more memory-intensive, but less cpu-intensive than Predictor. MPPC is the Microsoft Point-to-Point Compression and should only be used when Microsoft clients are connecting to the router.

```
r5(config-if)# compress ?
  mppc       MPPC compression type
  predictor  predictor compression type
  stac       stac compression algorithm
  <cr>
```

Compression should not be used if your CPU frequently exceed 65%. It also shouldn't be used if most of your data is already compressed. Once you enable compression you will need to force the interface to reset its Link Control Protocol for the compression to take effect. The quickest way is to use the `clear interface <interface>` global command.

If you have a lot of small TCP packets, you may want to enable TCP header compression. You can also control the maximum number of compressed connections as shown below.

```
r5(config-if)# ip tcp ?
  compression-connections  Maximum number of compressed connections
  header-compression       Enable TCP header compression

r5(config-if)# ip tcp header-compression ?
  passive  Compress only for destinations which send compressed headers
  <cr>
```

The passive option for header compression simply means that the router will only compress TCP packets if the incoming packets it receives on that same interface are also compressed. If the passive option is not used then all TCP packets are compressed.

MULTILINK

This is an example of a configuration task that you do not want to start configuring until you verify that before you make any new changes that your ISDN still dials. Was the last time you checked your ISDN dial when you did your basic configuration? Have you pinged across the circuit since you changed the CHAP hostname for example? Frequently candidates are in a hurry and do not continue to verify their configuration are working. Now that you have verified that your ISDN still works, lets move on to multilink.

By default, you should notice that in a basic ISDN configuration, only one channel comes up. Multilink allows the second BRI channel to be used based on utilization. You can configure you BRI to bring up the second channel when the load is at a specific threshold. The available values are 1 to 255 with 255 being full loaded. So, if you were asked to bring up the second channel when the utilization on the first channel exceeds 50% you would use the value 128.

To configure PPP multilink you need to complete five tasks. Configure a similar configuration on both routers.

1. Make sure your ISDN can dial with a base ISDN configuration[3]
2. Configure a dialer map for each channel[4]
3. Include the LDN with SPID's (if SPID's are required by your ISDN switch)
4. Configure the dialer load-threshold
5. Configure PPP multilink

```
r5(config)# interface BRI0
r5(config-if)# dialer map ip 192.168.1.6 4930622
r5(config-if)# dialer map ip 192.168.1.6 4930623
r5(config-if)# dialer load-threshold 1 either
r5(config-if)# ppp multilink

r6(config)# interface BRI0
r6(config-if)# dialer map ip 192.168.1.5 4930624
r6(config-if)# dialer map ip 192.168.1.5 4930625
r6(config-if)# dialer load-threshold 1 either
r6(config-if)# ppp multilink
```

Verify that both channels come up using a `show ip interface brief` and that multilink is working properly. If the threshold is set too high, the second channel will not come up. Set the threshold to "1" for testing purposes. Do not forget to change it back after you're done testing!

[3] Most of our ISDN examples use the same base ISDN configuration. Appendix A – CCIE Security configuration sheet contains basic configurations for the common topics. Additionally, the section on ISDN and Routing protocols later in this chapter also includes a base configuration for ISDN. Configurations such as those that use a dial string, one-way authentication, and PAP authentication are not what we consider a base configuration.

[4] This may contradict what you have read on Cisco's website and documentation. According to their documentation, you only need a single dialer map. We tested this on multiple IOS versions and the second channel did not come up with a single dialer map. The moment we added the second dialer map and ran an extended ping our second channel came up. This may be version dependent.

```
r6# show ip interface brief
Interface                  IP-Address      OK? Method Status
    Protocol
BRI0                       192.168.1.6     YES NVRAM  up                    up
BRI0:1                     unassigned      YES unset  up                    up
BRI0:2                     unassigned      YES unset  up                    up
Ethernet0                  unassigned      YES NVRAM  administratively down
    down
Serial0                    unassigned      YES NVRAM  administratively down
    down
Serial1                    unassigned      YES NVRAM  administratively down
    down
Virtual-Access1            unassigned      YES TFTP   up                    up
```

```
r6# show interface virtual-access 1
Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  MTU 1500 bytes, BW 128 Kbit, DLY 100000 usec,
     reliability 255/255, txload 3/255, rxload 3/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  Time to interface disconnect:  idle 00:01:55
  LCP Open, multilink Open
  Open: IPCP, CDPCP
  Last input 00:00:04, output never, output hang never
  Last clearing of "show interface" counters 00:04:42
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 2000 bits/sec, 1 packets/sec
  5 minute output rate 2000 bits/sec, 1 packets/sec
     28 packets input, 36140 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     29 packets output, 36120 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

CALLBACK

It may be cost effective to have a router call back another router when it receives an incoming call. For example, a hub router may call back all spoke routers when they want to establish a connection. A savvy administrator may negotiate a good rate with the ISDN provider for a large number of calls to be made from the hub location and a very cheap rate for the few, short calls placed by the spoke routers. Let's assume that R5 is the hub router and R6 is a spoke router. We want R5 to disconnect any calls it receives from R6 and then call back to R6. First, configure a basic ISDN configuration on both sides as shown in the example below.

```
hostname R5
!
username R6 password cisco
!
interface BRI0
 ip address 138.1.56.5 255.255.255.252
 encapsulation ppp
 dialer map ip 138.1.56.6 name R6 broadcast 4930622
 dialer-group 1
 ppp authentication chap


hostname R6
```

```
!
username R5 password cisco
!
interface BRI0
 ip address 138.1.56.6 255.255.255.252
 encapsulation ppp
 dialer map ip 138.1.56.5 name R5 broadcast 4930624
 dialer-group 1
 ppp authentication chap
```

To configure callback, let's start with the spoke router. The only command needed on the spoke is ppp callback request.

```
r6# conf t
r6(config)# interface BRI0
r6(config-if)# ppp callback request
```

Configuring the callback server is a little more involved. You MUST create a map-class in addition to the callback accept command. Don't forget to apply the map-class to the dialer-map. So, the first step is the simple ppp callback accept.

```
r5# conf t
r5(config)# interface BRI0
r5(config-if)# ppp callback accept
```

The second step is to create the map-class and apply it to the dialer-map. The name "CALLBACK" is arbitrary. The map-class can be used to either identify the caller by username or by dial-string. The username option should be sufficient for the lab exam.

```
r5(config)# map-class dialer CALLBACK
r5(config-map-class)# dialer callback-server username
```

Once you create the map-class, apply it to the dialer map. Be careful about the syntax as it can be picky about the order of the options.

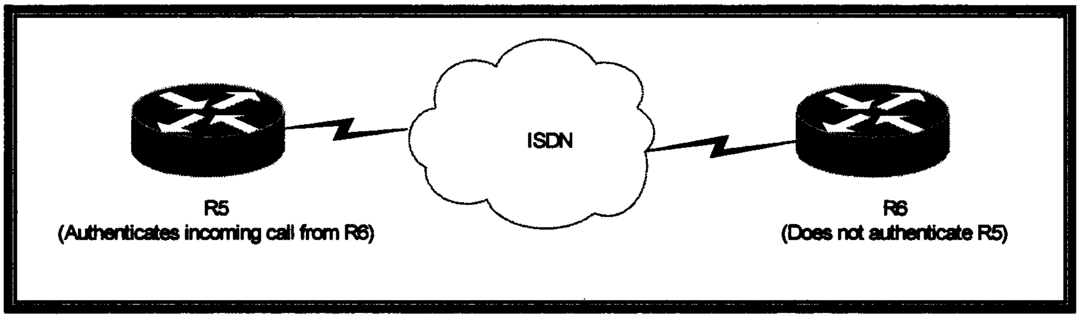
```
r5(config)# interface BRI0
r5(config-if)# dialer map ip 138.1.56.6 name R6 class CALLBACK broadcast
    4930622
```

UNIDIRECTIONAL AUTHENTICATION

By default, authentication is performed bi-directional. It may be desirable to just have one router perform authentication. This may be seen in a hub and spoke topology. The spokes do not require the hub to be authenticated since they already know that the router they are calling is valid. It is not possible to spoof an ISDN dial string. However, the hub router should always authenticate incoming callers in order to help prevent an unauthorized router from connecting to the network. All ISDN routers should authenticate incoming calls for security purposes.

In our example, R5 is the hub router. R6 is the remote site router. R6 is configured to authenticate incoming calls only. R5 requires no additional configuration since it performs a normal CHAP authentication. Be careful not to get this backwards! To make sure this works, simply use `debug ppp authentication` prior to configuring the unidirectional authentication. Observe the CHAP challenge and response. Once you configure unidirectional authentication, R6 no longer shows any CHAP authentication.

Figure 3.2 *Unidirectional authentication*


ISDN
R5
(Authenticates incoming call from R6)
R6
(Does not authenticate R5)

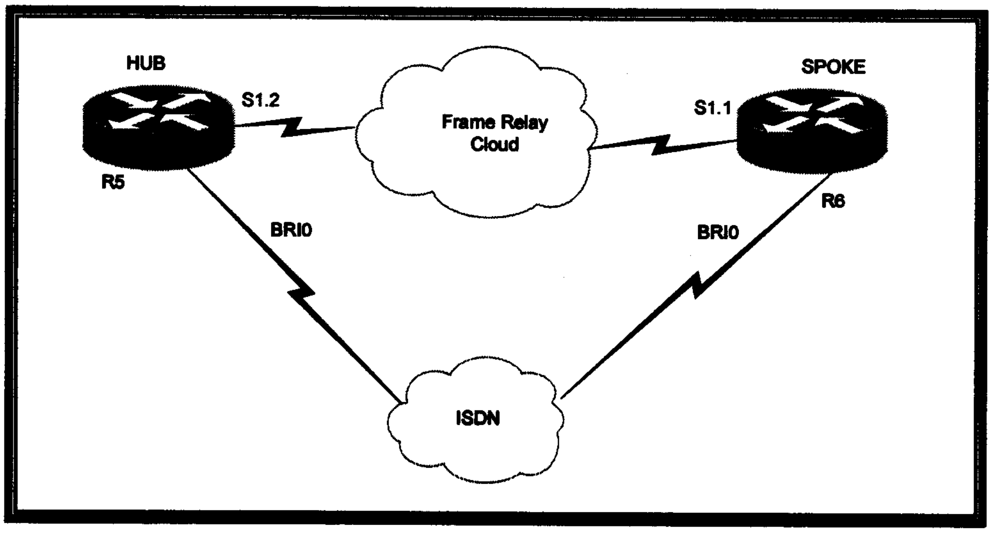
```
r6(config)# interface BRI0
r6(config-if)# ppp authentication chap callin

r6# debug ppp authentication
PPP authentication debugging is on
r6# debug ppp multilink events
Multilink events debugging is on

<Before we enabled unidirectional authentication>

00:16:51: BR0:1 PPP: Treating connection as a callout
00:16:51: BR0:1 CHAP: O CHALLENGE id 4 len 23 from "r6"
00:16:51: BR0:1 CHAP: I CHALLENGE id 4 len 23 from "r5"
00:16:51: BR0:1 CHAP: O RESPONSE id 4 len 23 from "r6"
00:16:51: BR0:1 CHAP: I SUCCESS id 4 len 4
00:16:51: BR0:1 CHAP: I RESPONSE id 4 len 23 from "r5"
00:16:51: BR0:1 CHAP: O SUCCESS id 4 len 4
00:16:51: BR0:1 MLP: Request add link to bundle
00:16:51: BR0:1 MLP: Adding link to bundle
00:16:51: Vi1 MLP: Added to huntgroup BR0
00:16:51: Vi1 MLP: Clone from BR0
00:16:51: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
00:16:51: Vi1 PPP: Treating connection as a callout
00:16:51: Vi1 MLP: Added first link BR0:1 to bundle r5
00:16:51: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 4930624 r5
00:16:52: %LINK-3-UPDOWN: Interface BRI0:2, changed state to up
00:16:52: BR0:2 PPP: Treating connection as a callout
00:16:52: BR0:2 CHAP: O CHALLENGE id 4 len 23 from "r6"
00:16:52: BR0:2 CHAP: I CHALLENGE id 4 len 23 from "r5"
00:16:52: BR0:2 CHAP: O RESPONSE id 4 len 23 from "r6"
00:16:52: BR0:2 CHAP: I SUCCESS id 4 len 4
00:16:52: BR0:2 CHAP: I RESPONSE id 4 len 23 from "r5"
00:16:52: BR0:2 CHAP: O SUCCESS id 4 len 4
00:16:52: BR0:2 MLP: Request add link to bundle
00:16:52: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed
    state to up
00:16:52: BR0:2 MLP: Adding link to bundle
00:16:52: Vi1 MLP: Added link BR0:2 to bundle r5
00:16:52: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
    changed state to up
00:16:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:2, changed
    state to up
00:16:58: %ISDN-6-CONNECT: Interface BRI0:2 is now connected to 4930625 r5

<After we enabled unidirectional authentication>

00:24:26: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
00:24:26: BR0:1 PPP: Treating connection as a callout
00:24:26: BR0:1 MLP: Request add link to bundle
00:24:26: BR0:1 MLP: Adding link to bundle
00:24:26: Vi1 MLP: Added to huntgroup BR0
```

```
00:24:26: Vi1 MLP: Clone from BR0
00:24:26: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
00:24:26: Vi1 PPP: Treating connection as a callout
00:24:26: Vi1 MLP: Added first link BR0:1 to bundle r5
00:24:26: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 4930624 r5
00:24:27: %LINK-3-UPDOWN: Interface BRI0:2, changed state to up
00:24:27: BR0:2 PPP: Treating connection as a callout
00:24:27: BR0:2 MLP: Request add link to bundle
00:24:27: BR0:2 MLP: Adding link to bundle
00:24:27: Vi1 MLP: Added link BR0:2 to bundle r5
00:24:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed
    state to up
00:24:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
    changed state to up
00:24:28: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:2, changed
    state to up
00:24:33: %ISDN-6-CONNECT: Interface BRI0:2 is now connected to 4930625 r5
```

Dial Backup (Interface Failure)

Dial backup is probably the easiest of all ISDN dial scenarios to understand and configure. It is probably one of the most common as well. Dial backup tracks a circuit such as a serial or frame relay link. If there is a failure at layer 1 or layer 2, the BRI interface will dial. The limitations here should be obvious. If there is a problem at layer 3, the BRI interface will not dial. Also, if you administratively shut down the interface being backed up, the BRI will not dial. To test ISDN backup, either disconnect the cable or make a change on the frame switch such as shutting down the interface.

Figure 3.3 *Serial interface backup using ISDN BRI interface*



There are three prerequisites to configure dial backup:

1. The BRI interfaces need to be configured properly prior to configuring backup. Make sure you can ping across your ISDN link and cause a dial.

2. There must be interesting traffic to force a dial. Backup by itself does not automatically cause a dial if there is a failure. A routing protocol or static route must be configured to use the ISDN after the serial interface failure.
3. Make sure the BRI interface is not in use. Both BRI channels must be unused while the interface is functioning as a backup interface.

Note Only one B channel can be used with legacy DDR and dial backup. You must configure Dialer Profiles in order to use both B channels.

The first step is to identify the interface you want to backup. As shown in Figure 3.3, we want to backup the point-to-point frame-relay interface between R5 and R6. R5 is the hub router and R6 is a spoke. We want R6 to dial the hub whenever it sees a failure on the frame-relay interface. The first number in the backup delay is how long it will wait before it dials. The second number is how long it will wait after the primary interface (serial 1.1 in this example) is up before it shuts down the BRI interface. The reason you may want it to stay up for a period of time after the serial has been restored is because of flapping. If your serial interface is flapping every 15 seconds you want to make sure your ISDN stays up. This way your traffic is not interrupted. The default for both of these values is 0. In our example, we want the ISDN to wait 10 seconds before coming up. This way if there is a brief outage, the ISDN won't dial unnecessarily. Then we want it to stay up for an additional 30 seconds after the serial comes up. If our serial is flapping every few seconds, our ISDN will stay up. Otherwise, it would flap too!

```
r6(config)# interface Serial1.1 point-to-point
r6(config-if)# backup interface BRI0
r6(config-if)# backup delay 10 30
```

Dial Backup (performance)

Dial backup can also be used to increase bandwidth thereby increasing network throughput. It can be configured to dial when a predefined utilization is reached. For example, if our serial link reaches 75% utilization, we can cause the ISDN to dial and help transmit the heavy load. Once the traffic (the total utilization on both the serial and ISDN) falls below 25% of the primary interface's configured bandwidth, the ISDN will disconnect. Both percentages are based on the configured bandwidth of the primary interface. You cannot configure backup load on a subinterface.

```
r6(config)# interface Serial1
r6(config-if)# backup interface BRI0
r6(config-if)# backup load 75 25
```

TIP Make sure you configure the bandwidth on the link if the default is not correct. By default, serial interfaces have a bandwidth of 1.544 Mbps.

Dialer Profiles

Dialer profiles allow a logical interface to use several available physical BRI interfaces. This way it has a pool of interfaces to choose from. Dialer profiles can be highly scalable and used for complex dialup topologies. To configure dialer profiles, you need to configure both the physical BRI interface as well as a logical dialer interface.

First, configure the physical interface with the required commands. The default encapsulation, HDLC, can be used instead of PPP and does not require that the encapsulation be configured. If PPP is selected and authentication is desired, it must be configured on the physical interface as well as the logical dialer interface. Each interface that participates in a dial profile must be a member of the dialer pool.

Below is an example of a router that has two BRI interfaces. Each interface can belong to the same pool or different pools. An interface can also belong to multiple pools simultaneously.

```
r6(config)# interface BRI0
r6(config-if)# encapsulation ppp
r6(config-if)# ppp authentication chap
r6(config-if)# dialer pool-member 1

r6(config)# interface BRI1
r6(config-if)# encapsulation ppp
r6(config-if)# ppp authentication chap
r6(config-if)# dialer pool-member 2
```

Next, configure the dialer interface. The dialer interface must have the same encapsulation and authentication configuration as the physical interfaces.

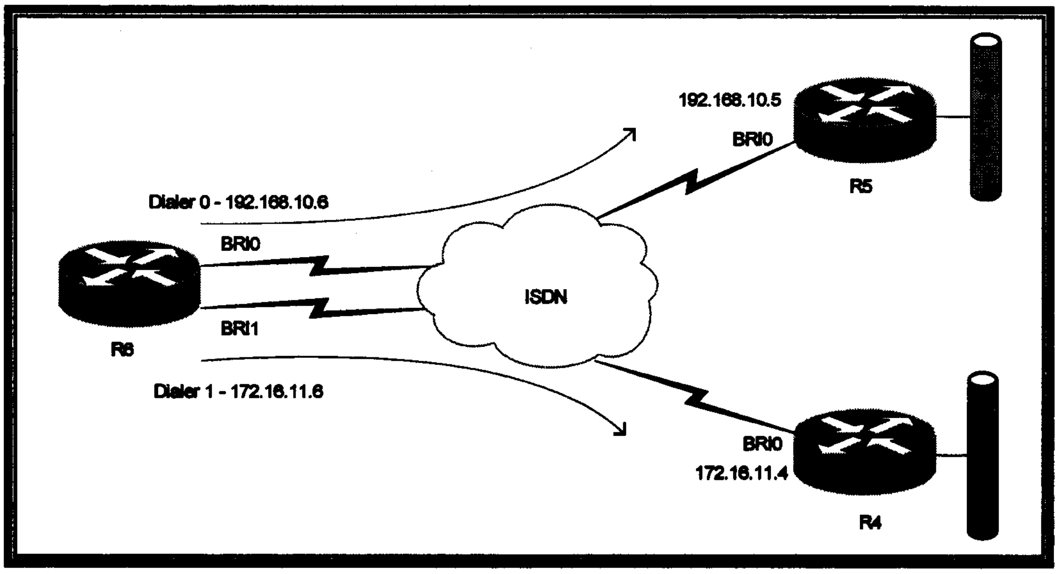
```
r6(config)# interface dialer0
r6(config-if)# encapsulation ppp
r6(config-if)# ppp authentication chap
r6(config-if)# ip address 192.168.10.6 255.255.255.0
r6(config-if)# dialer string 4930624
r6(config-if)# dialer remote-name r5
r6(config-if)# dialer pool 1
r6(config-if)# dialer-group 1
r6(config)# dialer-list protocol ip permit
```

If asked to configure dialer profiles it is very likely you will have to know some of the more common options. These options make dialer profiles very scalable and flexible.

Example 1 – Multiple next hop routers

Here we have a classic case of a router that needs to dial different numbers depending on the network that needs to be reached. R6 must dial R5 in order to reach network 192.168.1.0 and dial R4 to reach network 172.16.1.0. If there is an interface or circuit failure with either BRI0 or BRI1, we want to make sure we can still connect to both R5 and R4. Do not forget you need a route so the router knows which dialer interface to use!

Figure 3.4 *Multiple next hop routers*



```
r6(config)# interface BRI0
r6(config-if)# dialer pool-member 1
r6(config-if)# dialer pool-member 2

r6(config)# interface BRI1
r6(config-if)# dialer pool-member 1
r6(config-if)# dialer pool-member 2

r6(config)# interface dialer0
r6(config-if)# ip address 192.168.10.6 255.255.255.0
r6(config-if)# encapsulation ppp
r6(config-if)# dialer pool 1
r6(config-if)# dialer remote-name r5
r6(config-if)# dialer string 4930624
r6(config-if)# dialer-group 1
r6(config-if)# ppp authentication chap

r6(config)# interface dialer1
r6(config-if)# ip address 172.16.11.6 255.255.255.0
r6(config-if)# encapsulation ppp
r6(config-if)# dialer pool 2
r6(config-if)# dialer remote-name r4
r6(config-if)# dialer string 4930624
r6(config-if)# dialer-group 1
r6(config-if)# ppp authentication chap

r6(config)# ip route 192.168.1.0 255.255.255.0 dialer0
r6(config)# ip route 172.16.1.0 255.255.255.0 dialer1
```

Example 2 – Changing the Pool-Member Priority, Minimum and Maximum Channels

This example builds on Example 1 in that we have adjusted the priorities and the channels available to the dialer interface. Both BRI interfaces can be used for either dialer profile. BRI0 should have a higher priority for dialer interface 0 and therefore be the preferred interface when making a call. BRI1 should have a higher priority for dialer interface 1. Also, we are setting the minimum and maximum number of B channels available from the physical interface. In our configuration, we reserve a minimum of one B channel for each pool. The maximum is set at two, but remember that a BRI only has two B channels. This setup makes a little more sense when working with a PRI that has 23 B channels.

```
r6(config)# interface BRI0
r6(config-if)# dialer pool-member 1 priority 100 min-link 1 max-link 2
r6(config-if)# dialer pool-member 2 priority 50 min-link 1 max-link 2

r6(config)# interface BRI1
r6(config-if)# dialer pool-member 2 priority 100 min-link 1 max-link 2
r6(config-if)# dialer pool-member 1 priority 50 min-link 1 max-link 2
```

Example 3 – Map Classes

Map-classes allow you to configure several options and apply them to one or more dialer profiles. They can also be used with dialer-maps for legacy DDR. The word "DIALER1" is arbitrary. It is used to describe the map-class. Use any word that will help describe your map-class. The map-class can also be reused by other BRI or Dialer interfaces. A map-class allows you to configure the same settings on multiple dialer profiles without having to explicitly configure each parameter individually.

```
r6(config)# map-class dialer DIALER1
r6(config-map-class)# dialer ?
  callback-server        Enable callback return call
  enable-timeout         Set length of time an interface stays down before it
                         is available for dialing
  fast-idle              Set idle time before disconnecting line with an
                         unusually high level of contention
  idle-timeout           Set idle time before disconnecting line
  isdn                   ISDN Settings
  outgoing               Dial using the selected ISDN dialing plan.
  voice-call             Dial the configured number as a voice call
  wait-for-carrier-time  How long the router will wait for carrier

r6(config-map-class)# dialer idle-timeout 90
r6(config-map-class)# dialer fast-idle 10
r6(config-map-class)# dialer wait-for-carrier-time 5

r6(config)# interface dialer0
r6(config-if)# dialer string 4930624 map-class DIALER1
```

ISDN and Routing Protocols

Now that we've learned many of the ways to cause ISDN to dial, we need to exchange routing protocol information over the ISDN link. The problem with routing protocols is that they frequently bring up an ISDN line for periodic routing updates or because of hello packets. This behavior will either keep the ISDN line up indefinitely or bring it up very frequently typically resulting in very high toll charges. Therefore, we have to know how to use routing protocols over ISDN without dialing more than necessary. Many of the routing protocols have their own answer to DDR. Alternatively, we can use static routes and avoid using routing protocols completely although static routes are likely to be prohibited by the conditions of the lab exam.

All of these configuration examples were tested after starting with a base ISDN configuration and verifying we could successfully ping across the ISDN circuit. Below is the base configuration used for this section.

ROUTER 5

```
interface BRI0
  ip address 192.168.10.5 255.255.255.0
  encapsulation ppp
         dialer map ip 192.168.10.6 name r6 broadcast 4930622
  dialer map ip 192.168.10.6 name r6 broadcast 4930623
  dialer-group 1
  isdn switch-type basic-net3
  ppp authentication chap
dialer-list 1 protocol ip permit
```

ROUTER 6

```
interface BRI0
  ip address 192.168.10.6 255.255.255.0
  encapsulation ppp
  dialer map ip 192.168.10.5 name r5 broadcast 4930624
  dialer map ip 192.168.10.5 name r5 broadcast 4930625
  dialer-group 1
  isdn switch-type basic-net3
  ppp authentication chap
dialer-list 1 protocol ip permit
```

FLOATING STATIC ROUTES

Floating static routes allow us to avoid using routing protocols on ISDN links. The advantage of using floating static routes is that they are routing protocol independent and they are very simple to configure. A floating static route is simply a static route with an administrative distance higher than the routing protocols used in your network. Typically, you will see floating static routes configured with a value of 250 through 255. Remember that by default, a static route has an administrative distance of 0 or 1 depending on how you configure the route.

If you point the route to a connected interface, the route will be directly connected with a distance of 0. In the following example, we are pointing the route to the dialer0 interface. These static route configurations will work with legacy DDR or dialer profiles.

```
r5(config)# ip route 0.0.0.0 0.0.0.0 dialer0

r5# show ip route
02:05:12: %SYS-5-CONFIG_I: Configured from console by console
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, Dialer0
C       192.168.10.6/32 is directly connected, Dialer0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
S*   0.0.0.0/0 is directly connected, Dialer0
```

If you point the route to an IP address of a next hop router, the route will show as static with a distance of 1.

```
r5(config)# ip route 0.0.0.0 0.0.0.0 192.168.10.6

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is 192.168.10.6 to network 0.0.0.0

     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, Dialer0
C       192.168.10.6/32 is directly connected, Dialer0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
S*   0.0.0.0/0 [1/0] via 192.168.10.6
```

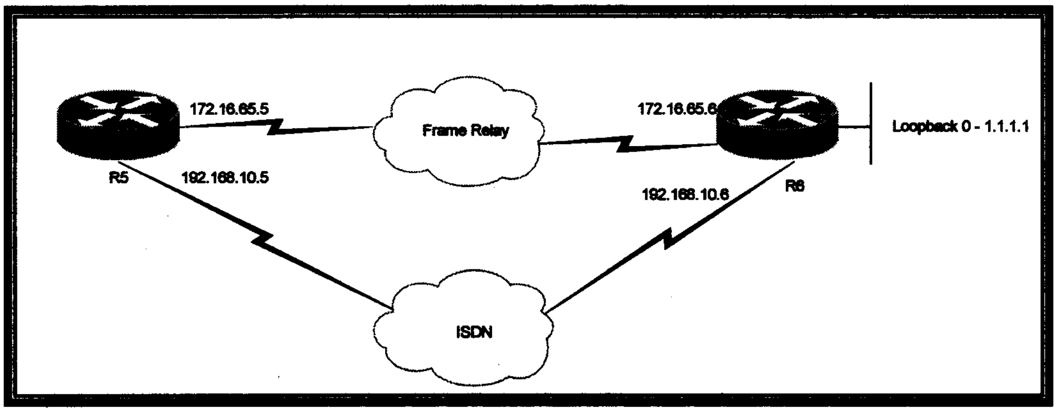
```
A floating static route is simply a static route with a high administrative
    distance.
```

```
r5(config)# ip route 0.0.0.0 0.0.0.0 192.168.10.6 254

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR
Gateway of last resort is 192.168.10.6 to network 0.0.0.0

     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, Dialer0
C       192.168.10.6/32 is directly connected, Dialer0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
S*   0.0.0.0/0 [254/0] via 192.168.10.6
```

OSPF DEMAND CIRCUIT

OSPF demand circuit is a topic all CCIE candidates should understand and know how to configure. With demand circuit, OSPF neighbors form an adjacency and then the BRI circuit is disconnected. It also suppresses hello packets that would normally bring up the circuit. OSPF will keep the circuit down by "spoofing" until it is needed. Make sure you configure it on only one side of the BRI circuit. It is not necessary to configure demand circuit on both ends of a BRI

Figure 3.5 *OSPF demand circuit topology*



When configuring demand circuit the most common mistake is forgetting the **broadcast** statement in the dialer map.

```
r5(config)# interface BRI0
r5(config-if)# dialer map ip 192.168.10.6 broadcast 4930622
r5(config-if)# dialer map ip 192.168.10.6 broadcast 4930623
r5(config-if)# ip ospf demand-circuit
```

OSPF Demand Circuit Verification

To verify your demand circuit is configured properly, shut down the frame-relay interface and check that the ISDN circuit dials and receives routes. It should already have an adjacency established from your initial demand circuit configuration. The command **show ip ospf interface** details OSPF information for the interface you select. This command shows us that demand circuit is configured and an adjacency is formed.

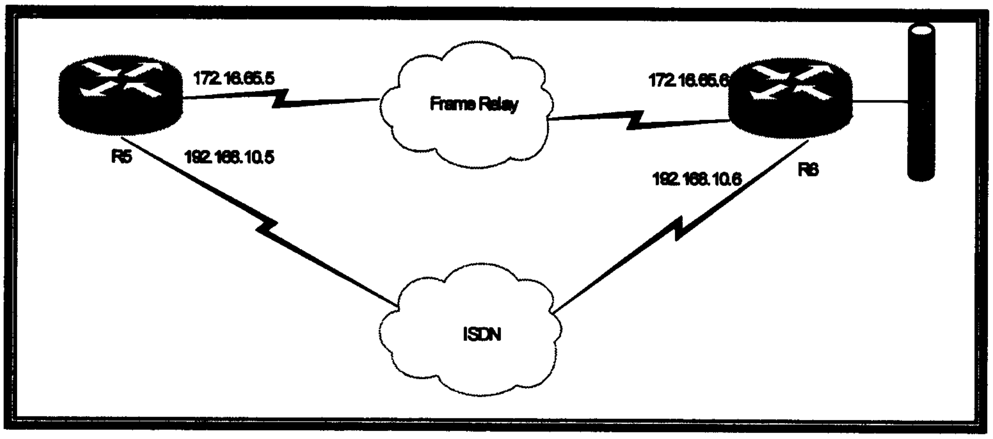
```
r5# show ip ospf interface bri0
BRI0 is up, line protocol is up (spoofing)
  Internet Address 192.168.10.5/24, Area 0
  Process ID 1, Router ID 192.168.10.5, Network Type POINT_TO_POINT, Cost:
   1562
  Configured as demand circuit.
  Run as demand circuit.
  DoNotAge LSA allowed.
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1  (Hello suppressed)
  Suppress hello for 1 neighbor(s)

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set
     1.0.0.0/32 is subnetted, 1 subnets
```

```
O IA     1.1.1.1 [110/1563] via 192.168.10.6, 00:01:25, BRI0
     192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.10.0/24 is directly connected, BRI0
C       192.168.10.6/32 is directly connected, BRI0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
O       172.16.65.0 [110/1626] via 192.168.10.6, 00:01:26, BRI0
```

Dialer Watch

Dialer watch can be used with any routing protocol. It looks for specific routes (configured by the administrator) to be present in the routing table. If a route it is "watching" for disappears, it will cause a dial. Dialer watch is particularly useful for EIGRP since there is no demand circuit or snapshot routing capability.

Figure 3.6 *Dialer watch topology*



As with any advanced ISDN configuration, verify that your ISDN dials and connects prior to configuring dialer watch. EIGRP is configured for all the networks shown in Figure 3.6.

Configuring Dialer Watch

Step 1 Identify which network(s) you want to "watch." Based on our topology, we want to watch the 10.10.10.0 network. Make sure that this exact network and mask is already in your routing table.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.10.0/24 is directly connected, BRI0
```

```
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
D       10.10.10.0 [90/2195456] via 172.16.65.6, 00:02:29, Serial1.2
```

Step 2 Configure a dialer map to point to the network you are watching. This is required because once our frame-relay connection goes down, there won't be a route to the 10.10.10.0 network. Configuring a dialer-map prevents us from having to configure a static route (which is probably prohibited by the CCIE lab instructions).

```
r5(config)# interface bri0
r5(config-if)# dialer map ip 10.10.10.0 broadcast 4930622
```

Step 3 Configure a dialer watch list. This is the network(s) that will be watched once it is applied to the interface.

```
r5(config)# dialer watch-list 1 ip 10.10.10.0 255.255.255.0
```

Step 4 Apply the watch-list to the BRI interface. This is similar to configuring a dialer-list and dialer-group. Notice a pattern?

```
r5(config)# interface bri0
r5(config-if)# dialer watch-group 1
```

Step 5 Prevent EIGRP from bringing the ISDN line up. We want the dialer watch event to bring up the ISDN circuit not EIGRP hello packets. Configure EIGRP as uninteresting traffic using an access list. Apply the access-list to the dialer-list. EIGRP cannot bring up the ISDN circuit, but any other IP traffic can.

```
r5(config)# access-list 100 deny eigrp any any
r5(config)# access-list 100 permit ip any any

r5(config)# dialer-list 1 protocol ip list 100
```

Dialer Watch Verification

To verify our dialer watch is configured properly using **debug** and **show** commands. Once we finish configuring dialer watch, we should see the router checking if the watched network is up. This happens at every idle-timeout interval. Remember this in case you are asked to have dialer watch check at an interval that is not default.

The first command is a basic `show interface`. We want to verify that the line protocol is up (spoofing).

```
r5# show interface bri0
BRI0 is up, line protocol is up (spoofing)
  Hardware is BRI
  Internet address is 192.168.10.5/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  Last input 00:00:06, output 00:00:06, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
     1458 packets input, 6157 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     1478 packets output, 6291 bytes, 0 underruns
     0 output errors, 0 collisions, 9 interface resets
     0 output buffer failures, 0 output buffers swapped out
     7 carrier transitions
```

Then check to make sure the route shows up via the expected interface. In our example, we expect the 10.10.10.0 network to be learned via the serial 1.2 interface when the network is functioning properly.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.10.0/24 is directly connected, BRI0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
D       10.10.10.0 [90/2195456] via 172.16.65.6, 00:39:31, Serial1.2
```

Before we test our dialer watch, run a **debug dialer events** so we can observe the dialer watch taking over.

```
r5# debug dialer events
Dial on demand events debugging is on

<Here we disconnect the serial interface>

03:06:25: DDR: Dialer Watch: watch-group = 1
03:06:25: DDR:    network 10.10.10.0/255.255.255.0 DOWN,
03:06:25: DDR:    primary DOWN
03:06:25: DDR: Dialer Watch: Dial Reason: Primary of group 1 DOWN
03:06:25: DDR: Dialer Watch: watch-group = 1,
03:06:25: DDR:    dialing secondary by dialer map 10.10.10.0 on BR0
03:06:25: BR0 DDR: Attempting to dial 4930622
03:06:26: DDR: Dialer Watch: watch-group = 1
03:06:26: DDR:    network 10.10.10.0/255.255.255.0 DOWN,
03:06:26: DDR:    primary DOWN
03:06:26: DDR: Dialer Watch: Dial Reason: Primary of group 1 DOWN
03:06:26: DDR: Dialer Watch: watch-group = 1,
03:06:26: %LINK-3-UPDOWN: Interface BRI0:1, changed state to up
03:06:26: BR0:1 DDR: Dialer Watch: resetting call in progress
03:06:26: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 4930622
03:06:26: BR0:1 DDR: dialer protocol up
03:06:27: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed
    state to up
03:06:32: %ISDN-6-CONNECT: Interface BRI0:1 is now connected to 4930622 r6
```

Now when we do a **show ip route** we see the 10.10.10.0 network is being learned via the BRI0 interface.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```

```
        i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
        U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.10.0/24 is directly connected, BRI0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
D       172.16.65.0 [90/41024000] via 192.168.10.6, 00:01:22, BRI0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.10.10.0/24 [90/40537600] via 192.168.10.6, 00:01:22, BRI0
C       10.10.10.0/32 is directly connected, BRI0
```

When we restore the serial 1.2 interface, the dialer watch discovers the route is back up on the next idle timeout period.

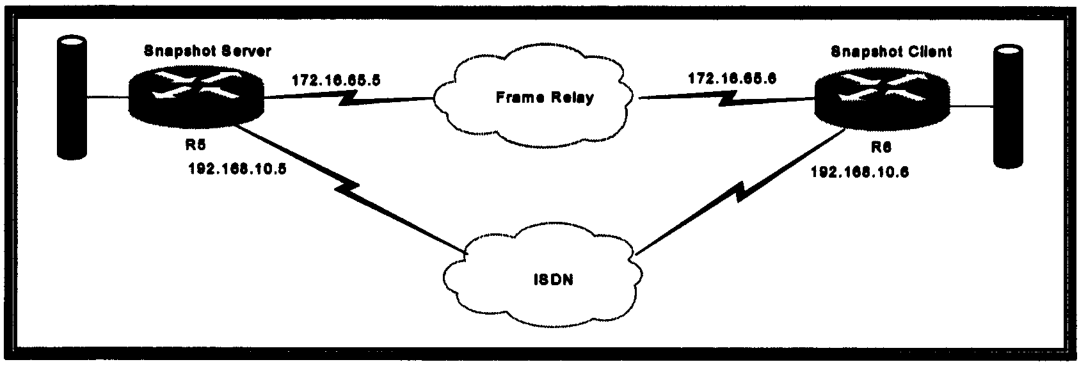
```
04:15:56: BR0:1 DDR: idle timeout
04:15:56: DDR: Dialer Watch: watch-group = 1
04:15:56: DDR:    network 10.10.10.0/255.255.255.0 UP,
04:15:56: DDR:    primary UP
04:15:56: BR0:1 DDR: disconnecting call
04:15:56: BR0:1 DDR: Dialer Watch: resetting call in progress
04:15:56: DDR: Dialer Watch: watch-group = 1
04:15:56: DDR:    network 10.10.10.0/255.255.255.0 UP,
04:15:56: DDR:    primary UP
04:15:56: %ISDN-6-DISCONNECT: Interface BRI0:1  disconnected from 4930622 r6,
    call lasted 30 seconds
04:15:56:  isdn_Call_disconnect()

04:15:56: %LINK-3-UPDOWN: Interface BRI0:1, changed state to down
04:15:56: BR0:1 DDR: disconnecting call
04:15:56: DDR: Dialer Watch: watch-group = 1
04:15:56: DDR:    network 10.10.10.0/255.255.255.0 UP,
04:15:56: DDR:    primary UP
04:15:57: %LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed
    state to down
```

SNAPSHOT ROUTING

Snapshot routing works with distance vector routing protocols such as RIP and IGRP. It works by taking a "snapshot" of the current routing table and storing it for long periods of time. Snapshot routing avoids the constant updates of most distance vector routing protocols.

Snapshot routing is client and server based. The client is usually a remote office while the server is typically a headquarters or main office. The client(s) call the server periodically to request routing updates. This call happens at much less frequent intervals than a typical routing update.

Figure 3.7 *Snapshot routing topology*



RIP is configured and routing for the networks shown in Figure 3.7. A base ISDN configuration is already in place described in the begging of this section.

The server configuration is very simple. You only need one command (assuming your base configuration is working properly). The number "5" designates the active time. When the client calls the server, the server determines how long the line will stay up for routing updates. In our example, we set it at 5 minutes. The word "dialer" tells the snapshot server to use dialer maps to reach the client.

```
r5(config)# interface bri0
r5(config-if)# snapshot server 5 dialer
```

The client configuration is a little more complex. The client requires an additional dialer map to support snapshot routing. The number "1" signifies a sequence number. This sequence number is necessary to support multiple snapshot servers. If the client needs to call more than one server, this sequence number becomes significant. The snapshot client command has two values: active period and quiet period. The active period must match the snapshot server. The active period is the amount of time the circuit is kept up in order to exchange routing updates. Once the active timer expires, the client does not call the server again until the quiet period expires which is 60 minutes in our example. However, interesting traffic will bring up the line based on your dialer-list configuration. These values are only used to bring up the line for routing updates. Once the interface is up, all traffic is permitted unless denied by an access-list and access-group. The dialer option tells the snapshot client to use the dialer maps to reach the server.

```
r5(config)# interface bri0
r6(config-if)# dialer map snapshot 1 broadcast 4930624
r6(config-if)# snapshot client 5 60 dialer
```

When interesting traffic brings up the ISDN circuit, the routers will also exchange routing updates in addition to transmitting user traffic. This may not be desirable since you may want the full bandwidth available for user traffic. Use the `suppress-statechange-update` option to prevent the snapshot routers from exchanging updates during this active time.

```
r6(config-if)# snapshot client 5 60 suppress-statechange-update dialer
```

Snapshot Routing Verification

Now we need to verify our configuration. First, use the show snapshot command to verify that snapshot has the correct settings. You should notice the first set of examples is during an active period. The second set is during a quiet period.

```
r5# show snapshot bri0
BRI0 is up, line protocol is upSnapshot server
  Options: dialer support
  Length of active period:         5 minutes
   For ip address: 192.168.10.6
    Current state: active, remaining time: 3 minutes
    Connected dialer interface:
       BRI0:1

r6# show snapshot bri0
BRI0 is up, line protocol is upSnapshot client
  Options: dialer support, stay asleep on carrier up
  Length of active period:         5 minutes
  Length of quiet period:          60 minutes
  Length of retry period:          8 minutes
   For dialer address 1
    Current state: active, remaining/exchange time: 2/2 minutes
    Connected dialer interface:
       BRI0:1
    Updates received this cycle: ip

r5# show snapshot bri0
BRI0 is up, line protocol is upSnapshot server
  Options: dialer support
  Length of active period:          5 minutes
   For ip address: 192.168.10.6
    Current state: server post active, remaining time: 2 minutes

r6# show snapshot bri0
BRI0 is up, line protocol is upSnapshot client
  Options: dialer support, stay asleep on carrier up
  Length of active period:          5 minutes
  Length of quiet period:           60 minutes
  Length of retry period:           8 minutes
   For dialer address 1
    Current state: quiet, remaining: 59 minutes
```

When our frame-relay is connected, we see the 50.0.0.0 route coming across the serial 1.1 frame-relay interface.

```
r6# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
R    50.0.0.0/8 [120/8576] via 172.16.65.5, 00:00:03, Serial1.1
C    192.168.10.0/24 is directly connected, BRI0
     172.16.0.0/24 is subnetted, 2 subnets
R       172.16.0.0 [120/10476] via 172.16.65.5, 00:00:03, Serial1.1
C       172.16.65.0 is directly connected, Serial1.1
     60.0.0.0/24 is subnetted, 1 subnets
C       60.60.60.0 is directly connected, Ethernet0
```

Next, we want to disconnect the frame-relay circuit and make sure we can still reach the 50.0.0.0 network from R6. Remember that IGRP converges very slowly, so be patient. Since we aren't patient we cleared the routing table and forced it to converge.

```
r6# clear ip route *

r6# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
I    50.0.0.0/8 [100/158350] via 192.168.10.5, 00:03:04, BRI0
C    192.168.10.0/24 is directly connected, BRI0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.65.0 is directly connected, Serial1.1
```

```
     60.0.0.0/24 is subnetted, 1 subnets
C       60.60.60.0 is directly connected, Ethernet0
r6# ping 50.50.50.50

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 50.50.50.50, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/32 ms
```

TYPICAL GOTCHAS!

- Setting the wrong switch type
- Not setting the isdn switch-type, either globally or on the interface
- Forgetting to add broadcast to your dialer maps
- Missing local authentication to authenticate the incoming call
- Remembering that the authentication must match the name and password that the opposite side is
    - using
- Missing the username and password for CHAP or PAP
- Remembering that PAP must be told what username and password to send (CHAP automatically
    - send its hostname and knows the password it expects to receive)
- Configuring the dialer map with the correct phone number (not calling yourself)
- Missing the dialer-list or dialer-group
- Forgetting to add the map class to your dialer map statement on the callback server
- Not adding a dialer map statement for the network you are watching with dialer watch

CHAPTER 4

BRIDGING

Bridges and switches are data communications devices that operate principally at Layer 2 of the OSI reference model. There are 4 types of bridges: transparent, source-route, translational, and source-route transparent. The current format of the CCIE lab only tests transparent bridging.

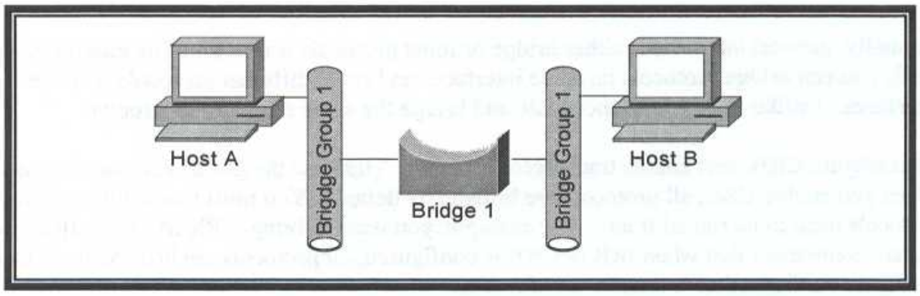
Transparent bridging (TB)

Transparent bridges, as the name implies, operate transparently to network hosts. A transparent bridge learns the network's topology by looking at the source MAC address of incoming frames from all attached networks. For example, a bridge sees a frame arrive on port 1 from Host A, the bridge now knows that Host A can be reached through the segment connected to port 1. The bridge creates an entry in its forwarding table for Host A. This process is known as "learning."

When a frame is received on one of the bridge's interfaces, the bridge looks up the frame's destination address in its forwarding table. If the table contains an entry for the destination address, the frame is forwarded out the indicated port. If no entry is found, the frame is flooded to all ports except the inbound port. The process of sending the frame out all ports is known as "flooding."

To configure TB you need only two commands. First, you need to choose which bridge protocol to use (DEC, IBM, IEEE). IBM is only used for source-route bridging. Then enable the bridge group on the interface(s). The following example allows us to bridge traffic on R8 between the two Ethernet networks.

Figure 4.1 *Transparent Bridging*

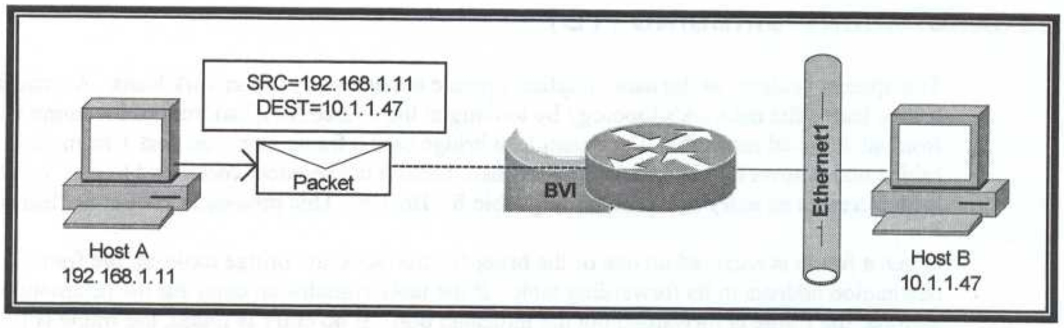


```
r8(config)# bridge 1 protocol ieee
r8(config)# interface ethernet 0
r8(config-if)# bridge-group 1
r8(config)# interface ethernet 1
r8(config-if)# bridge-group 1
```

Integrated Routing and Bridging (IRB)

IRB allows the router to bridge and route the same protocol on a router. In order to use IRB, create a Bridged Virtual Interface (BVI). After the BVI is configured, the router can send routable protocols that were bridged to the BVI to be routed. For example, an IP packet arrives on a router's bridged interface. The destination is out another interface that is not configured for routing instead of bridging. With IRB, the router then sends the packet to the appropriate interface to be routed. The protocols that you want the BVI to route must be explicitly configured.

Figure 4.4 *Integrated Routing and Bridging*



```
r1(config)# interface ethernet 0
r1(config-if)# bridge-group 1

r1(config)# interface ethernet 1
r1(config-if)# ip address 10.1.1.1 255.255.255.0

r1(config)# bridge irb
r1(config)# interface bvi1
r1(config-if)# bridge 1 route ip
```

Concurrent Routing and Bridging (CRB)

Normally, networking devices either bridge or route protocols across all of its interfaces. With CRB, you can bridge protocols on some interfaces and route different protocols on other interfaces. Unlike IRB, you cannot route and bridge the same protocol on a router.

To configure CRB, first enable transparent bridging. Then use the global command **bridge crb**. When you enable CRB, all protocols are bridged by default. You must manually configure which protocols need to be routed if any. For example, you want to bridge IPX and route IP on the same router. Remember that when IRB or CRB is configured, all protocols are bridged by default. That is why no configuration was necessary for IPX.

```
r1(config)# interface ethernet 0
r1(config-if)# bridge-group 1
r1(config)# interface ethernet 1
r1(config-if)# ip address 10.1.1.1 255.255.255.0
r1(config)# bridge crb
r1(config)# bridge 1 route ip
```

TYPICAL GOTCHAS!

- Forgetting to define the spanning tree protocol that Transparent Bridging will use
- Not specifying which traffic you want to be routed for IRB or CRB
- Forgetting that the bridge-group number and the BVI number must match in IRB

CHAPTER 5

CATALYST 3550 SWITCHING

SPANNING TREE (STA)

STA was designed to allow for redundant paths between bridges while preventing loops. For example, two ports connect Bridge 1 and Bridge 2. In order to prevent a loop only one port is in use at a time. The port not in use is put into a status known as "blocking." If there is a failure with the primary port, the blocked port will become active.

Each port in every bridge also is assigned a unique identifier, which is typically its own MAC address. Each switch port is associated with a path cost, which represents the cost of transmitting a frame onto a LAN through that port. Path costs have a default value depending on the type of port, but they can be changed manually by network administrators. Path costs on Catalyst switches are determined by interface type (Fast Ethernet, Gigabit Ethernet, Token-ring, FDDI, etc.).

The spanning-tree calculation occurs when the bridge is powered up and whenever a topology change is detected. The calculation involves sending configuration messages between bridges. These communication messages are known as bridge protocol data units, or BPDU's. BPDU's contain information identifying the bridge that is presumed to be the root and the distance from the sending bridge to the root bridge (root path cost). They also contain the bridge and port identifier of the sending bridge, as well as the age of the information.

Bridges exchange BPDU's at regular intervals (typically one to four seconds). If there is a failure of some sort and neighboring bridges stop receiving BPDU's they will initiate a spanning-tree recalculation.

SPANNING TREE CONFIGURATION

The Catalyst 3550 has many of the same STP features of the set-based switches with some enhancements and notable differences. Since the 3550 is IOS based, any changes made that you want to be permanent will need to be saved using `write mem` or `copy run start`. This rule also applies to the VLAN database covered later in this chapter.

DEFAULT STP VALUES

It may be helpful to know the default values for STP. Table 5.1 illustrates these values as of software version 12.1.4.

Table 5.1 *STP Values*

Feature	Default Setting
Switch priority	32768
STP and VLAN Port costs	1000 Mbps : 4 100 Mbps: 19 10 Mbps: 100
Port priority	128
Hello time	2 seconds
Forward-delay time	15 seconds
Maximum-aging time	20 seconds
Port Fast, BPDU Guard, UplinkFast, BackboneFast, Root guard	Disabled

Disable STP

Although unlikely, there may be situations where it is desirable to disable spanning-tree. If there is a loop in your layer 2 topology disabling spanning-tree will result in data being lost and severe network degradation. It is almost never recommended to disable spanning-tree. Use this command with caution.

To disable STP on a VLAN use the global command **no spanning-tree vlan <vlan_id>.**

Root Switch

The 3550 maintains a separate STP instance for each VLAN. So you will need to configure each VLAN individually that you want to make root. To make a switch root for a VLAN use the **spanning-tree vlan <vlan_id> root primary.** There are two optional parameters: diameter and hello-time. Diameter sets the maximum number of switches between any two hosts. This helps prevent loops by stopping a frame from endlessly traveling in a loop. Once the maximum number of switches is traversed the frame is dropped. Hello-time is a value between 1 and 10 seconds. It is simply how often a switch will send a "hello" frame to a switch that it has a trunk connection. The hello-time affects how quickly spanning-tree will detect a fault and use a new path.

```
Switch(config)# spanning-tree vlan 1 root primary diameter 3 hello-time 5
```

Secondary Root Switch

The secondary root switch is useful in situations where you have two switches that are typically in parallel (usually redundant core switches) where one should backup the other as the root bridge. The only difference between setting the primary and secondary root is the option after root.

```
Switch(config)# spanning-tree vlan 1 root secondary diameter 3 hello-time 5
```

Port Priority

When a switch has multiple paths it will put one in forwarding and the rest will be blocking. This is used for loop prevention. If all the interfaces have the same priority value (this will always happen unless you explicitly configure the port's value), then STP will use the lowest numbered interface (for example, it would use interface fastethernet 0/1 before fastethernet 0/2 assuming they were both connected to the root bridge) as the forwarding port. The only interfaces that can be configured for port priority are physical interfaces (not VLAN interfaces) and port-channel logical interfaces.

Note The default for IEEE STP is 128. The value can be from 0 to 255. The lower the number the higher the priority.

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# spanning-tree port-priority 1
```

Path Cost

By default, path costs are determined by the media speed of the interface. It is typically not necessary to change this value. Table 5-1 illustrates the path costs for each of the interface types available on the 3550.

The same basic rules apply to path cost as they do to port priority. If there are two paths from a switch to the root bridge, the first consideration is priority. If priority is equal, then path cost will be the deciding factor. If they are all equal, the switch will add the port priority and port ID of both interfaces. STP will then disable the link with the lowest value.

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# spanning-tree vlan 1 cost 10
```

Switch Priority for a VLAN

This command is simply changing the likelihood of the switch becoming root for a particular VLAN. The lower the number, the more likely the switch will be the root bridge.

```
Switch(config)# spanning-tree vlan 1 priority 8192
```

Timers

There are several timers that can affect how quickly STP detects and corrects a spanning-tree issues. Typical issues include loops, ports going down, a new root bridge being introduced in to the network, etc. Sometimes these issues are planned behavior and other times they are a result of a network problem. Bringing a new core switch online to be the root bridge would be considered a planned temporary outage. The core switch going down and an access layer switch becoming the root bridge for a VLAN is very likely to cause problems least of which would likely be suboptimal paths.

Hello Time

The hello time is the interval between configuration messages sent by the root switch. It is recommended that you use the `spanning-tree vlan <vlan_id> root` command to change the hello timer.

```
Switch(config)# spanning-tree vlan 1 hello-time 5
```

Forwarding-Delay Time

The forward delay is how many seconds the switch will wait to change a port from learning and listening to the forwarding state. This range for this timer is 4 to 30 seconds with 15 being the default for IEEE. Lowering this number will speed up the time it takes spanning-tree to be able to use a port to forward frames. However, if this number is too low and there are network problems, there may be additional instability.

```
Switch(config)# spanning-tree vlan 1 forward-time 10
```

Maximum-Aging Time

The maximum-aging time is the number of seconds the switch will wait for a STP configuration message before it will attempt to reconfigure. The range for this timer is 6 to 40 seconds with 20 being the default for IEEE.

```
Switch(config)# spanning-tree vlan 1 max-age 15
```

Port Fast

Access ports that do not participate in spanning-tree, such as those with end stations (typically workstations or servers) attached to them, can immediately begin forwarding and skip the listening and learning states. It also prevents the switch from generating Topology Change Notifications (TCN) to other switches thereby reducing traffic.

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# spanning-tree portfast
```

BPDU Guard

A port configured for Port Fast should not receive BPDU's. If it does, the switch by default will put the port in to blocking state. BPDU Guard will instead shut the port down. This helps prevent topology loops.

```
Switch(config)# spanning-tree portfast bpduguard
```

Root Guard

Root Guard is very similar to BPDU Guard. If a switch configured with Root Guard detects BPDU's on a port configured for Port Fast it will allow that device to participate in STP and will only block the port if the other switch is attempting to become the root bridge.

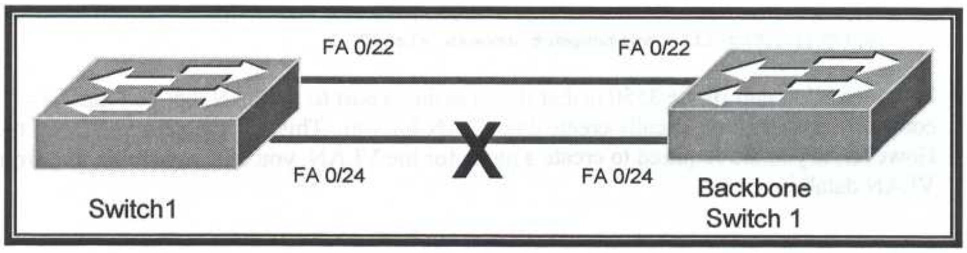
```
Switch(config)# interface fastethernet0/1
Switch(config-if)# spanning-tree guard root
```

Load Sharing Using STP

In this example, the Backbone switch is the root bridge. Switch 1 selected FA 0/22 as the forwarding port to the root bridge. If you don't know you need to go back and read from the beginning of the chapter!

Let's assume that we have 10 VLAN's each switch. What if we want traffic for VLAN's 1-5 to use port FA 0/22 and and VLAN's 6-10 to use FA 0/24. This could be very helpful if a single 100Mbps port is not sufficient bandwidth.

Figure 5.1 *Spanning-Tree Load Sharing*



Since they both have the same media speeds the best approach would be to change the port priority for each VLAN. Remember the default cost is 128.

```
Switch1(config)# interface fastethernet0/24
Switch1(config-if)# spanning-tree vlan 6 cost 10
Switch1(config-if)# spanning-tree vlan 7 cost 10
Switch1(config-if)# spanning-tree vlan 8 cost 10
Switch1(config-if)# spanning-tree vlan 9 cost 10
Switch1(config-if)# spanning-tree vlan 10 cost 10
```

VIRTUAL LAN'S (VLAN)

VLANs are logical subnets or segments. Unlike a physical subnet, the devices do not need to share the same physical cable segment. Devices can be connected to different switches in different locations and still be part of the same subnet.

VLAN CONFIGURATION

The 3550 is very different from the set-based Catalyst switches. The VLAN database is stored as a completely separate file from the rest of the configuration. The file is stored in NVRAM as vlan.dat.

To enter VLAN configuration mode type `vlan database`. This command is not entered from config mode. If you do not specify a name, the switch will automatically create one based on the VLAN number.

```
Switch# vlan database
Switch(vlan)# vlan 2
VLAN 2 added:
    Name:  VLAN0002

Switch# vlan database
Switch(vlan)# vlan 3 name VLAN_C
VLAN 3 added:
    Name:  VLAN_C
```

To delete a VLAN, type `no vlan <number>` from the VLAN database.

```
Switch# vlan database
Switch(vlan)# no vlan 3
Deleting VLAN 3…
```

Once the VLAN's are created, they need to be applied to the appropriate interfaces. To make port fa 0/2 to be part of VLAN 2 enter the switchport access command as shown.

```
Switch(config)# interface fastethernet0/2

Switch(config-if)# switchport access vlan 2
```

The interesting part of the 3550 is that if you assign a port to a VLAN that you did not yet configure, it will automatically create this VLAN for you. This may be a time saver on the lab. However, if you are required to create a name for the VLAN, you will have to do that from the VLAN database.

```
Switch(config)# interface fastethernet0/4
Switch(config-if)# switchport access vlan 4
Access VLAN does not exist.  Creating vlan 4
```

TRUNKING

Trunking carries multiple VLANs from one switch to another. You can use either IEEE 802.1Q or Cisco's proprietary Inter-Switch Link (ISL). Note that certain switches may be limited in which versions they support. The Catalyst 3550 series supports both ISL and 802.1Q.

Aside from ISL being proprietary, the main difference between them is that 802.1Q inserts a VLAN identifier in to the frame header. This is called frame tagging.

There are 4 steps to configure a trunk.

1. Set the trunk mode
2. Set the trunk encapsulation
3. Set the native VLAN (required for 802.1Q)
4. Set the default VLAN

TRUNK MODE

To configure a trunk mode use the `switchport mode` command. A port configured for with a regular VLAN is permanently nontrunking even if it is connected to a port on another switch that is configured for trunking. For example, assume the two switches below are connected together via port fa0/2. If Switch1 is configured for a VLAN and the backbone switch is configured as a trunk, Switch1's fa0/2 will still not become a trunk.

```
Switch1(config)# interface fastethernet0/2
Switch1(config-if)# switchport access vlan 2

Backbone(config)# interface fastethernet0/2
Backbone(config-if)# switchport access mode trunk
```

An interface configured for desirable mode will become a trunk port if the other end is set to trunk, desirable or auto mode. An interface configured for auto mode will become a trunk if the other end is set to trunk or desirable. An interface configured for trunk mode will become a trunk regardless of the other end's configuration. However, both sides need to be working for the trunk to actually work. The table below shows the different modes and if they will become a trunk.

Table 5.2 *Trunking modes*

Switch 1	Switch 2	Trunk ?
Desirable	Desirable	Yes
Desirable	Auto	Yes
Desirable	Trunk	Yes
Auto	Desirable	Yes
Auto	Trunk	Yes
Auto	Auto	No

The syntax to configure the 3 types of trunks is shown below. You may only have one trunk mode on an interface.

```
Switch1(config-if)# switchport mode dynamic desirable
Switch1(config-if)# switchport mode dynamic auto
Switch1(config-if)# switchport mode trunk
```

Trunk Encapsulation

The 3550 supports both ISL and 802.1Q. You must configure the same encapsulation type on both ends of a trunk. Remember that 802.1Q trunks require a native VLAN. The native VLAN must be the same on both ends of an 802.1Q trunk. A trunk port can also be configured to negotiate with the other switch to determine the encapsulation type used.

```
Switch1(config)# interface fastethernet0/22
Switch1(config-if)# switchport trunk mode trunk
Switch1(config-if)# switchport trunk encapsulation isl
```

-OR-

```
Switch1(config)# interface fastethernet0/23
Switch1(config-if)# switchport trunk mode trunk
Switch1(config-if)# switchport trunk encapsulation dot1q
Switch1(config-if)# switchport trunk native vlan 1
```

-OR-

```
Switch1(config)# interface fastethernet0/24
Switch1(config-if)# switchport trunk mode trunk
Switch1(config-if)# switchport trunk encapsulation negotiate
```

VLAN Restrictions

It may be desirable to limit which VLAN's can traverse a trunk. By default, a trunk will allow all VLAN's. Default VLAN's may not be removed from a trunk.

VLAN's can be added or removed from the allowed list one at a time or in groups. You can also tell a trunk to allow all or none of the VLAN's with a single keyword. The first example permits VLAN's 1-10 and 15. It demonstrates the syntax of using hyphens and commas. Make sure you are comfortable with this syntax as it may save you time.

```
Switch1(config-if)# switchport trunk allowed vlan add 1-10,15

Switch1(config-if)# switchport trunk allowed vlan remove 8
```

```
Switch1(config-if)# switchport trunk allowed vlan all

Switch1(config-if)# switchport trunk allowed vlan none
```

EtherChannel

EtherChannel allows you to bind between two and eight interfaces to increase the bandwidth between two switches. There are several requirements that must be met prior to configuring Etherchannel. The following parameters should be the same on all ports in the channel group.

1. Speed
2. Duplex
3. Native VLAN (if using 802.1Q)
4. Allowed-VLAN list
5. STP path cost for each VLAN[5]
6. STP port priority for each VLAN
7. STP Port Fast setting

The 3550 has the ability to use Port Aggregation Protocol (PAgP) to negotiate the capabilities of each interface. PAgP will dynamically add or remove interfaces from the channel group. It also adds the channel group to spanning-tree as a single port.

It is usually recommended not to disable PAgP. PAgP can be disabled by configuring "on" mode as discussed in the next section.

Table 5.3 *Default EtherChannel Settings*

Feature	Default Setting
Channel groups	None
Layer 3 port-channel logical interface	None
PAgP mode	Auto and silent
PAgP learn method	Aggregate-port learning
PAgP priority	128
Load balancing	Based on source MAC address

Layer 2 Channels

To configure a layer 2 channel, first verify all the settings are the same on all the interfaces. Next configure the interfaces as access or trunk ports. Then configure the channel-group. There are several mode types that are similar to trunking. These include auto, desirable, and on. Auto and desirable also have the ability to use non-silent mode. The default mode is auto with silent. Silent mode is used when connecting to servers or packet analyzers.

```
Switch1(config)# interface fastethernet0/24
Switch1(config-if)# switchport trunk mode trunk
Switch1(config-if)# channel-group 1 mode desirable
```

[5] If there are different path costs the EtherChannel will still be compatible. It is recommended to match the path cost regardless of it not causing an incompatibility.

The channel modes are described as follows:

- Auto – enables PAgP only if it detects a PAgP compatible switch at the other end of the link. An interface set to auto will not send PAgP packets, but it will respond to them.
- Desriable – actively send PAgP packets and attempts to negotiate.
- On – interface will channel without PAgP. All interfaces must be configured for "on" to use this option.
- Non-silent – used to connect to other PAgP capable switches. Silent mode suppresses PAgP messages to devices that do not understand PAgP such as file servers.

Table 5.4 *Channel Group settings to result in Etherchannel*

Switch1	Backbone1	EtherChannel?
Desirable	Desirable	Yes
Desirable	Auto	Yes
Auto	Auto	No
On	Auto	No
On	Desirable	No

Layer 3 Channels

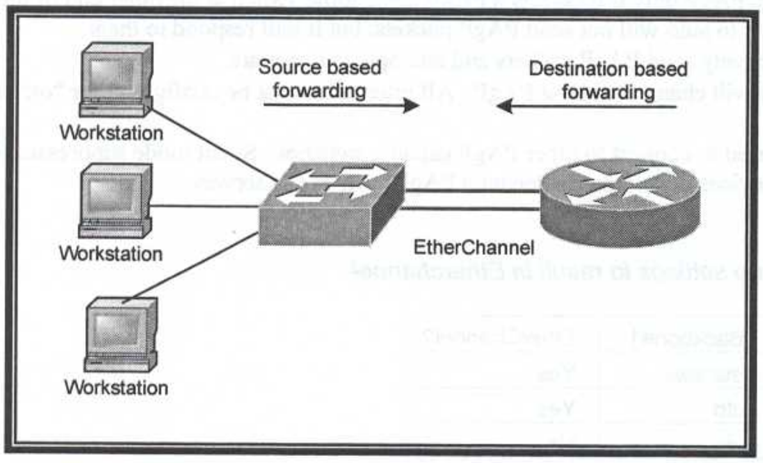
There are two steps to create a Layer 3 channel. The first step is to create the logical port-channel interface. The second step is to apply physical interfaces to the port-channel.

```
Switch1(config)# interface port-channel 1
Switch1(config-if)# no switchport
Switch1(config-if)# ip address 192.168.1.1 255.255.255.0

Switch1(config)# interface range fa0/22-24
Switch1(config-if)# no ip address
Switch1(config-if)# channel-group 1 mode desirable
```

Load Balancing

Since EtherChannel uses multiple physical interfaces, there must be some decision made on how to use these interfaces. By default, load balancing is based on the source MAC address. There may be instances where this is not the most optimal. For example, in Figure 5-2 we have several hosts communicating with a router. Communication from the router back to the workstations would all go over the same physical interface port resulting in suboptimal use of bandwidth. However, we want the communication from the workstations to the router to be source based. In this situation the most optimal forwarding is depending on the direction.

Figure 5.2 *EtherChannel Load Balancing*



```
Switch1(config)# port-channel load-balance src-mac
```

Router Port Channel

The setup shown in Figure 5.2 would also require the router to be configured for EtherChannel[6]. There are 3 main steps:

1. Create the logical port-channel interface
   a. Set the encapsulation
   b. Set the IP address
2. Configure the physical interfaces to be used in the channel
3. Apply the physical interfaces to the channel-group

```
router(config)# int port-channel 1
router(config-if)# full-duplex
```

-ISL Configuration-

```
router(config-if)# int port-channel 1.1
router(config-if)# encapsulation isl 1
router(config-if)# ip address 10.1.1.1 255.255.255.0

router(config-if)# int port-channel 1.2
router(config-if)# encapsulation isl 2
router(config-if)# ip address 10.2.2.1 255.255.255.0
```

-802.1Q Configuration-

```
router(config-if)# int port-channel 1.1
router(config-if)# encapsulation dot1q 1 native
router(config-if)# ip address 10.1.1.1 255.255.255.0

router(config-if)# int port-channel 1.2
router(config-if)# encapsulation dot1q 1 native
router(config-if)# ip address 10.2.2.1 255.255.255.0
```

-Physical Interface Configuration -

[6] These steps require a minimum of IOS version 12.1.3T

```
router(config-if)# int fa0/0
router(config-if)# speed 100
router(config-if)# full-duplex
router(config-if)# no shut
router(config-if)# channel-group 1

router(config-if)# int fa0/1
router(config-if)# speed 100
router(config-if)# full-duplex
router(config-if)# no shut
router(config-if)# channel-group 1
```

VLAN Trunking Protocol (VTP)

VTP helps maintain VLAN consistency throughout a network by preventing duplicate VLAN names, incorrect VLAN-type specifications, and security violations. When you configure VTP on multiple switches you can add, delete, or edit VLANs on one switch and have that information propagated to all the switches that are part of the VTP domain.

A switch participating in VTP can be configured for three different modes of operation: server, client, or transparent. The VTP server is where you add, delete, or edit VLAN information. The clients use the VTP messages they receive from the server to modify their VLAN configuration. You cannot make any VLAN changes on a client. Transparent switches do not participate in VTP at all, except they will forward received VTP advertisements out their trunk ports if version 2 is enabled.

VTP advertisements are sent out all ISL and 802.1Q trunks. If a switch receives a VTP advertisement that is using a different VTP domain or if has an earlier revision number the switch will ignore it. Revision numbers are incremented each time there is a change that affects the VTP database. This prevents a new switch from being introduced to the network and wiping out the VTP database of the switched network.

The only step required to configure VTP is to set the domain name. This name does not have anything to do with DNS. It is simply a management domain. There are several options for VTP including password, version, and pruning.

VTP Mode

In order to configure VTP, start with the switch or switches that you want to act as the server(s). Then configure client switches with the same VTP domain name and set the mode to client. By default, all switches are VTP servers.

```
Switch# vlan database
Switch(vlan)# vtp server
Device mode already VTP SERVER.
Switch(vlan)# vtp domain NETCG
Changing VTP domain name from NULL to NETCG
```

You can also disable VTP by setting the mode as transparent. If you choose not to use VTP, you will have to disable it with the command below. You do not need a VTP domain if you are using transparent mode.

```
Switch# vlan database
Switch(vlan)# vtp transparent
```

VTP Options

VTP has the ability to use passwords, prune VLAN's, and use version 1 or 2. All switches in a domain must use the same version. If you enable v2 on a server, all switches in the network that support v2 will automatically switch to using only v2. The advantages of using version 2 inlcude the following:

- Unrecognized Type-Length-Value (TLV) support – a switch will propogate configuration changes even if it cannot parse the TLV.

- Version-Dependent Transparent Mode – a transparent switch running VTP v1 will not forward messages if the domain name and version do not match. If it is running v2, it will forward VTP messages without checking the domain name and version.

- Consistency Checks – checks are not performed when information is received from another switch or read from NVRAM as long as the MD5 digest is correct. The check is run when new information is entered through the CLI, Cluster Management Software, or via SNMP.

To configure VTP version 2 mode, use the `vtp v2-mode` command.

```
Switch# vlan database
Switch(vlan)# vtp v2-mode
V2 mode enabled.
```

If you are going to use passwords, make sure the password is consistent throughout the VTP domain. The password can be anywhere from 8 to 64 characters.

```
Switch# vlan database
Switch(vlan)# vtp password ccie7146
Setting device VLAN database password to ccie7146.
```

Pruning prevents a switch from advertising VTP messages about a VLAN if the downstream switch does not have any ports assigned to that VLAN. In order for a VLAN to be pruned, you must configure the prune eligible list. Only VLAN's 2 – 1001 are prune eligible on trunk parts.

```
Switch# vlan database
Switch(vlan)# vtp pruning
```

Once pruning is enabled, allowing or denying VLAN's from being prune eligibile follows similar syntax as allowing VLAN's on a trunk. The options are add, remove, except, and none.

```
Switch(config)# interface fa0/24
Switch(config-if)# switchport trunk pruning vlan add 2-10,27
```

TYPICAL GOTCHAS!

- Forgetting that the LOWEST priority on a switch is preferred
- Not naming VLANs as specified
- Erasing the vlan.dat file on cat1 only to have VTP from cat2 replicate those same VLANs back to cat1
- Making sure the encapsulation on both sides of a trunk are the same
- Remembering to configure the native VLAN for 802.1Q trunking
  Not matching all port requirements for EtherChannel to work

Section II

Layer 3 Routing Protocols

Chapter 6	General Routing
Chapter 7	OSPF
Chapter 8	BGP
Chapter 9	EIGRP
Chapter 10	RIP
Chapter 11	RIP Version 2
Chapter 12	Redistribution

CHAPTER 6

GENERAL ROUTING

There are several commands and topics that are shared by most routing protocols. This section briefly touches on some of the major routing protocol independent commands. These topics are covered in detail throughout the routing protocol chapters.

NETWORK COMMAND

The network command is used to designate which networks a routing protocol will advertise. It also indicates which interfaces will send and receive routing updates and form neighbor relationships (such as OSPF and EIGRP). Each protocol has a slightly different behavior.

OSPF does not require that the network mask be exact. However, it must be large enough to capture the interface. For example, network 172.16.0.0 0.0.255.255 will place any interfaces (and their corresponding networks) that match 172.16.0.0 \16 in to OSPF.

BGP requires that the network mask be exact. If an exact route does not exist in the IP routing table, BGP will not advertise it with the network command. Exact routes include the directly connected interfaces as well.

RIP and IGRP do not require the exact mask since they only work on classful boundaries. They do not advertise subnet mask information. Even if you configure a specific network such as 10.10.10.0 /24, the router will automatically change your network statement in your configuration to 10.0.0.0.

PASSIVE INTERFACE

A passive interface prevents routing updates from being sent on a link but not received. It also prevents OSPF and IS-IS neighbor relationships from being formed. If an interface does not need to run a routing protocol use the **passive-interface** command to disable it for that particular interface.

```
r5(config)# router rip
r5(config-router)# passive-interface bri0
```

For example, R5 is configured for RIP. All of its interface are in the major 10.0.0.0 network. Once we enable RIP routing for the 10.0.0.0, RIP is running on all interfaces. However, we have an ISDN link that we do not want to run RIP on. One of our options is to use passive-interface to prevent RIP from sending routing update packets on that link thereby bringing it up.

CHAPTER 6

GENERAL ROUTING

There are several commands and topics that are shared by most routing protocols. This section briefly touches on some of the major routing protocol independent commands. These topics are covered in detail throughout the routing protocol chapters.

Network Command

The network command is used to designate which networks a routing protocol will advertise. It also indicates which interfaces will send and receive routing updates and form neighbor relationships (such as OSPF and EIGRP). Each protocol has a slightly different behavior.

OSPF does not require that the network mask be exact. However, it must be large enough to capture the interface. For example, network 172.16.0.0 0.0.255.255 will place any interfaces (and their corresponding networks) that match 172.16.0.0 \16 in to OSPF.

BGP requires that the network mask be exact. If an exact route does not exist in the IP routing table, BGP will not advertise it with the network command. Exact routes include the directly connected interfaces as well.

RIP and IGRP do not require the exact mask since they only work on classful boundaries. They do not advertise subnet mask information. Even if you configure a specific network such as 10.10.10.0 /24, the router will automatically change your network statement in your configuration to 10.0.0.0.

Passive Interface

A passive interface prevents routing updates from being sent on a link but not received. It also prevents OSPF and IS-IS neighbor relationships from being formed. If an interface does not need to run a routing protocol use the **passive-interface** command to disable it for that particular interface.

```
r5(config)# router rip
r5(config-router)# passive-interface bri0
```

For example, R5 is configured for RIP. All of its interface are in the major 10.0.0.0 network. Once we enable RIP routing for the 10.0.0.0, RIP is running on all interfaces. However, we have an ISDN link that we do not want to run RIP on. One of our options is to use passive-interface to prevent RIP from sending routing update packets on that link thereby bringing it up.

OSPF can be configured to assign a separate distance depending on the route type.

```
r5(config-router)# distance ospf ?
    external     External type 5 and type 7 routes
    inter-area   Inter-area routes
    intra-area   Intra-area routes

r5(config-router)# distance ospf external 200
```

OSPF (and all other routing protocols) can be configured to assign a distance based on matching an access-list. First, identify the network(s) you want to alter the distance. In our example, we change the 8.8.8.8 /32 network to have a distance of 200. As you can see from our `show ip route`, the current distance for this route is the OSPF standard of 110. Also, note the source of the routing update. In this example, it is 10.5.56.6.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
O IA    1.1.1.0 [110/65] via 172.16.0.1, 00:01:06, Serial1.1
     2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/65] via 172.16.0.2, 00:01:06, Serial1.1
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/11] via 10.5.56.6, 00:01:07, Ethernet0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.0.0/24 is directly connected, Serial1.1
O       172.16.0.1/32 [110/64] via 172.16.0.1, 00:01:07, Serial1.1
O       172.16.0.2/32 [110/64] via 172.16.0.2, 00:01:07, Serial1.1
C       172.16.65.0/24 is directly connected, Serial1.2
     8.0.0.0/32 is subnetted, 1 subnets
O IA    8.8.8.8 [110/75] via 10.5.56.6, 00:01:08, Ethernet0
     10.0.0.0/24 is subnetted, 3 subnets
O IA    10.8.8.0 [110/84] via 10.5.56.6, 00:01:08, Ethernet0
C       10.5.56.0 is directly connected, Ethernet0
O       10.5.68.0 [110/74] via 10.5.56.6, 00:01:08, Ethernet0
```

Once we've identified the network we want to change, simply create an access-list that will match this network.

```
r5(config)# access-list 1 permit 8.8.8.8 0.0.0.0
```

Then configure the distance on routing protocol updates received from neighbor 10.5.56.6 and that match access-list 1.

```
r5(config)# router ospf 1
r5(config-router)# distance 200 10.5.56.6 255.255.255.255 1
```

To verify our new distance has taken effect, we cleared the routing table and then used the `show ip route` command. In order to shorten the output we specifically asked for the routing table entry for the 8.8.8.8 network. The standard `show ip route` command would have provided the same information we are looking for. But, it helps to know these types of shortcuts.

```
r5# show ip route 8.8.8.8
Routing entry for 8.8.8.8/32
```

```
Known via "ospf 1", distance 200, metric 75, type inter area
Redistributing via ospf 1
Last update from 10.5.56.6 on Ethernet0, 00:00:04 ago
Routing Descriptor Blocks:
* 10.5.56.6, from 8.8.8.8, 00:00:04 ago, via Ethernet0
 Route metric is 75, traffic share count is 1
```

Table 6.1 *Administrative Distances*

Route Type	Administrative Distance
Connected	0
Static route pointing to an interface	0
Static route pointing to a next hop	1
EIGRP Summary Route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
Exterior Gateway Protocol	140
External EIGRP	170
Internal BGP	200
Unknown Route or Unreachable	255

TYPICAL GOTCHAS!

- Forgetting to specify the ‘mask’ in BGP network statements
- Not turning off split-horizon on a Multipoint sub-interface that’s running EIGRP

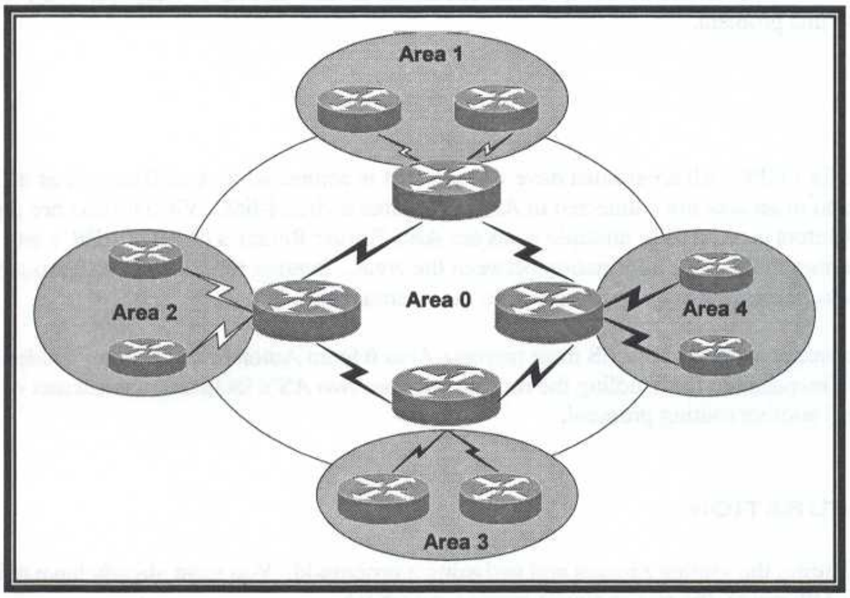
  Remembering that passive interfaces only disable sending, not receiving, of routing updates – except for routing protocols that rely on a neighbor relationship

CHAPTER 7

OPEN SHORTAGE PATH FIRST

OSPF is a link-state protocol based on Dijkstra's algorithm. An OSPF network must have a core area (area 0) and typically, there are one or more areas that are connected to the core. All areas must have an area border router that is directly connected to area 0. Routers within an area are known as internal routers. These routers exchange Link State Advertisements (LSA's) with each other in order to determine which networks are available and which link is the best path to a network. Internal routers do not communicate with any routers outside of the area. An organization typically has many OSPF routers that full under one administrative domain called an autonomous system.

Figure 7.1 *Basic OSPF Network*



OSPF Areas Types

- **Backbone (transit area)** - Always labeled area "0." It accepts all route types and is used to connect multiple areas. All other areas must connect to this area in order to exchange and route information. The only exception to this is the use of virtual-links. When interconnecting multiple areas, the backbone area is the core area to which all other areas must connect.

- **Area** - Accepts internal and external routes as well as summary information.

- **Stub** - Refers to an area that does not accept routes learned outside of the AS or from another routing protocol. If routers need to route to networks outside the autonomous system, they use a default route.

- **Not-so-stubby** – Also known as NSSA. It is the same as a stub area except it accepts external routes that are redistributed in to the NSSA. This is useful if you want to accept redistributed routes from another routing protocol.

- **Totally Stub** – All LSA's except Type 1 and 2 are blocked. Intra-area routes and the default route are the only routes passed within a totally stubby area. This is Cisco proprietary.

Peer Relationships

OSPF hello packet information must be the same on all routers in an area for neighbor relationships to be formed. A hello packet contains the following information:

- Hello/Dead Interval
- Area ID
- Authentication Password (if configured)
- Stub Area Flag

Hello packets are also only transmitted on broadcast capable networks. NBMA networks such as frame-relay present a problem for OSPF neighbor relationships. The upcoming section on Frame-relay and OSPF explains the various solutions to this problem.

Area 0

Area 0 is the core area for OSPF. All areas must have a router that is connected to Area 0 as well as its internal area. The only exception of an area not connected to Area 0 requires a virtual-link. Virtual-links are discussed later in this chapter. Routers connected to multiple areas are Area Border Router's (ABR). ABR's are responsible for maintaining the routing information between the areas. Internal routers receive all routes from the ABR except for those routes that are contained within the internal area.

Traffic destined for networks outside of the AS must traverse Area 0 to an Autonomous System Border Router (ASBR). The ASBR is responsible for handling the routing between two AS's including the Internet or when redistributing OSPF and another routing protocol.

Basic OSPF Configuration

Enable OSPF by configuring the routing process and assigning a process-id. You must already have at least 1 IP address configured. Otherwise, the router cannot create a router-ID.

```
router(config)# router ospf 1
```

Configure which networks are part of OSPF and what area they belong to. You can configure the wildcard mask to capture one network or multiple networks. Alternatively, you can configure a wildcard mask with a host address to place that interface in to OSPF. There is no functional difference between the two network commands shown below. Either command will enable OSPF on Ethernet 0.

```
r1(config)# interface ethernet0
r1(config-if)# ip address 192.168.1.1 255.255.255.0
```

```
r1(config)# router ospf 1
r1(config)# network 192.168.1.0 0.0.0.255 area 0
```

-OR-

```
r1(config)# network 192.168.1.1 0.0.0.0 area 0
```

It would be useful to use a less specific wildcard mask to save time in a large router. Imagine the situation below where you have five loopbacks in the same major network. Instead of entering five different network commands, you can enter a single network command.

```
r1(config)# interface loopback0
r1(config-if)# ip address 192.168.0.1 255.255.255.0
r1(config)# interface loopback1
r1(config-if)# ip address 192.168.1.1 255.255.255.0
r1(config)# interface loopback2
r1(config-if)# ip address 192.168.0.1 255.255.255.0
r1(config)# interface loopback3
r1(config-if)# ip address 192.168.3.1 255.255.255.0
r1(config)# interface loopback4
r1(config-if)# ip address 192.168.4.1 255.255.255.0

r1(config)# router ospf 1
r1(config-router)# network 192.168.0.0 0.0.255.255 area 0
```

Frame-Relay and OSPF

OSPF configuration over frame-relay has a few potential pitfalls. There are four main issues to remember when configuring OSPF over frame-relay.

- Make sure network types or interface types match
- Hello and Dead timers must match
- Broadcast, Non-Broadcast, and Point-to-Multipoint try to elect a DR/BDR
- Frame maps require the "broadcast" statement

OSPF determines the network type based on the frame-relay interface type. However, the network type can be changed on a per-interface basis. The following table illustrates the default network types, timers, and DR election properties for frame-relay interfaces.

Table 7.1 *OSPF and Frame-Relay Interfaces*

Interface Type	Network Type	Hello Timer	Dead Timer	DR Election?
Physical	NON_BROADCAST	30 seconds	120 seconds	Yes
Point-to-Point	POINT_TO_POINT	10 seconds	40 seconds	No
Point-to-Multipoint	NON_BROADCAST	30 seconds	120 seconds	Yes

So, if we have a mix of physical and point-to-multipoint interfaces we can establish an adjacency without additional configuration right? Wrong! Non-broadcast networks are exactly as it is titled: non-broadcast. This means they will not broadcast hello packets and establish neighbor relationships. In this situation, we have two choices:

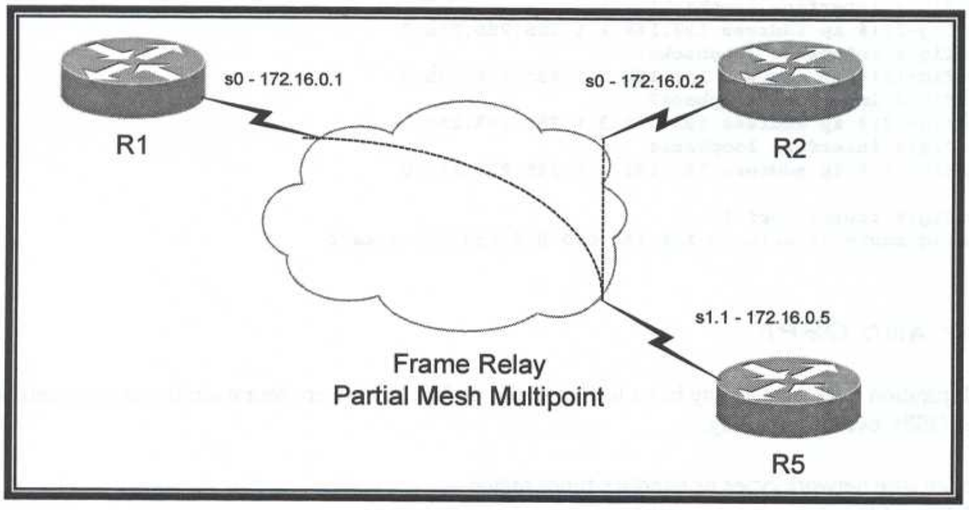
1. Configure all interfaces to be the same network type. In a hub and spoke environment, it is highly recommended to configure all interfaces for point-to-multipoint. This is the preferred method to configure OSPF on a hub and spoke frame-relay network.

2. Configure static neighbors. This is especially useful if you are prohibited from changing the network type.

OSPF over Frame-Relay Configuration

Figure 7.2 illustrates a typical OSPF hub and spoke topology. We will be using this topology to explain how to configure OSPF over frame-relay.

Figure 7.2 *Frame-relay and OSPF topology*



OSPF over Frame-relay configuration (changing network types)

To find out what your current network type is, use the **`show ip ospf interface`** command. This command is very useful when configuring and troubleshooting OSPF. It gives us valuable information such as area, router ID, network type, timers, DR state, neighbor status, etc. The first example is for a physical interface. The second example is for a point-to-multipoint interface.

```
r1# show ip ospf interface s0
Serial0 is up, line protocol is up
  Internet Address 172.16.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 172.16.0.1, Interface address 172.16.0.1
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:25
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 0, maximum is 0
  Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 0, Adjacent neighbor count is 0
    Suppress hello for 0 neighbor(s)

r5# show ip ospf interface s1.1
Serial1.1 is up, line protocol is up
  Internet Address 172.16.0.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 192.168.10.5, Interface address 172.16.0.5
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:28
  Neighbor Count is 0, Adjacent neighbor count is 0
  Suppress hello for 0 neighbor(s)
```

Since we cannot establish an adjacency over a non-broadcast network, we configured all three routers with a point-to-multipoint network type. We were then able to establish adjacencies.

```
r5(config)# interface Serial1.1
r5(config-if)# ip ospf network point-to-multipoint

r1(config)# interface Serial0
r1(config-if)# ip ospf network point-to-multipoint

r2(config)# interface Serial0
r2(config-if)# ip ospf network point-to-multipoint

r5# show ip ospf interface s1.1
Serial1.1 is up, line protocol is up
  Internet Address 172.16.0.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:26
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 172.16.0.2
    Adjacent with neighbor 172.16.0.1
  Suppress hello for 0 neighbor(s)
```

OSPF over Frame-Relay Configuration (Static Neighbors)

Neighbors can be statically configured if you do not have the option of changing the network type. This will work fine, but there is still an issue with the Designated Router (DR) and Backup Designated Router (BDR) election as explained in the next section.

```
r5(config)# router ospf 1
r5(config-router)# neighbor 172.16.0.1
r5(config-router)# neighbor 172.16.0.2

r1(config)# router ospf 1
r1(config-router)# neighbor 172.16.0.5

r2(config)# router ospf 1
r2(config-router)# neighbor 172.16.0.5
```

Designated and Backup Designated Router Elections

All OSPF networks (except point-to-point) need a designated router (DR). You can do this one of three ways:

1. Set the priority at the interface level
2. Manually configure the router ID
3. Configure the router with the highest loopback address

During the election process if the priority is not configured, the router with the highest Router ID is set as the DR. The router ID (if not manually configured) uses the highest loopback address. If no loopback address is configured, it uses the highest interface address.
Routing advertisements are sent to the DR only. The DR then distributes the routing updates to the rest of the routers in the area. This prevents the need for a full-mesh neighbor relationship to exchange routing updates. All routers on the network need only send their routing updates to one router instead of many. In a large network with many neighbors on a single network, this can save considerable bandwidth and processing.

To manually set a router as the DR or BDR for a network, configure the priority for the interface. By default, the priority is 1 for all routers.

```
r5(config)# interface Serial1.1
r5(config-if)# ip ospf priority 200
```

This is still an issue for frame-relay. If R5 reboots or the serial interface goes down, either R1 or R2 will take over as the DR. This will prevent routing from working properly on the network if R5 is not the DR since R1 and R2 are not directly connected. To solve this problem, prevent the spokes from ever becoming a DR or BDR. This is accomplished by setting an OSPF interface to priority to 0. A router with OSPF priority 0 is not eligible to participate in DR/BDR elections. If the state is "DROTHER", it is neither a DR nor BDR.

```
r1(config)# interface s0
r1(config-if)# ip ospf priority 0

r2(config)# interface s0
r2(config-if)# ip ospf priority 0

r5# show ip ospf interface s1.1
Serial1.1 is up, line protocol is up
  Internet Address 172.16.0.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 5.5.5.5, Interface address 172.16.0.5
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:25
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 2.2.2.2
    Adjacent with neighbor 1.1.1.1
  Suppress hello for 0 neighbor(s)

r1# show ip ospf interface s0
Serial0 is up, line protocol is up
  Internet Address 172.16.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type NON_BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State DROTHER, Priority 0
  Designated Router (ID) 5.5.5.5, Interface address 172.16.0.5
  No backup designated router on this network
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:26
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 5.5.5.5  (Designated Router)
  Suppress hello for 0 neighbor(s)
```

LOOPBACKS

Loopbacks are used extensively in the CCIE lab in order to create additional networks without having a network interface for each network.

OSPF advertises loopbacks with a host mask (/32) in the routing table. If the actual network mask should be advertised, there are two ways to accomplish this. The preferred way is to configure an **ip ospf network** on the interface. We recommend using a point-to-point interface. If your IOS version does not support changing the network type for a loopback, use the **area range** command.

```
r5(config)# interface loopback 0
r5(config-if)# ip ospf network point-to-multipoint
OSPF: Invalid type for interface

r5(config-if)# ip ospf network point-to-point
```

To verify that the route shows up with the true network mask, use the **show ip route** command on one of the other OSPF routers. The loopback from R2 shows up with a /32 host mask. The loopback from R5 shows up with the actual mask configured for that loopback interface.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter      area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
     2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/129] via 172.16.0.5, 01:16:00, Serial0
     5.0.0.0/24 is subnetted, 1 subnets
O IA    5.5.5.0 [110/65] via 172.16.0.5, 01:16:00, Serial0
     6.0.0.0/32 is subnetted, 1 subnets
O IA    6.6.6.6 [110/75] via 172.16.0.5, 01:16:00, Serial0
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       172.16.0.5/32 [110/64] via 172.16.0.5, 01:16:00, Serial0
C       172.16.0.0/24 is directly connected, Serial0
O       172.16.0.2/32 [110/128] via 172.16.0.5, 01:16:01, Serial0
     8.0.0.0/32 is subnetted, 1 subnets
O IA    8.8.8.8 [110/139] via 172.16.0.5, 01:16:02, Serial0
     10.0.0.0/24 is subnetted, 3 subnets
O IA    10.8.8.0 [110/148] via 172.16.0.5, 01:16:02, Serial0
O IA    10.5.56.0 [110/74] via 172.16.0.5, 01:16:02, Serial0
O IA    10.5.68.0 [110/138] via 172.16.0.5, 01:16:02, Serial0
```

-OR-

```
r1(config)# router ospf 1
r1(config-router)# area 1 range 1.1.1.0 255.255.255.0
```

Here we verify that R5 sees the 1.1.1.0 network with the proper mask.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
O IA    1.1.1.0 [110/65] via 172.16.0.1, 01:17:41, Serial1.1
     2.0.0.0/32 is subnetted, 1 subnets
O IA    2.2.2.2 [110/65] via 172.16.0.2, 01:20:24, Serial1.1
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     6.0.0.0/32 is subnetted, 1 subnets
O       6.6.6.6 [110/11] via 10.5.56.6, 01:20:24, Ethernet0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.0.0/24 is directly connected, Serial1.1
O       172.16.0.1/32 [110/64] via 172.16.0.1, 01:20:24, Serial1.1
O       172.16.0.2/32 [110/64] via 172.16.0.2, 01:20:24, Serial1.1
C       172.16.65.0/24 is directly connected, Serial1.2
     8.0.0.0/32 is subnetted, 1 subnets
O IA    8.8.8.8 [110/75] via 10.5.56.6, 01:20:27, Ethernet0
     10.0.0.0/24 is subnetted, 3 subnets
```

```
O IA    10.8.8.0 [110/84] via 10.5.56.6, 01:20:27, Ethernet0
C       10.5.56.0 is directly connected, Ethernet0
O       10.5.68.0 [110/74] via 10.5.56.6, 01:20:27, Ethernet0
```

Router ID

The Router ID identifies the OSPF router. Although it may have many interfaces running OSPF, it needs a single, unique ID to identify the router to the rest of the network. The router ID (if not manually configured) uses the highest loopback address. If no loopback address is configured, it uses the highest interface address. The router ID is typically not used in configuration commands except for virtual links.

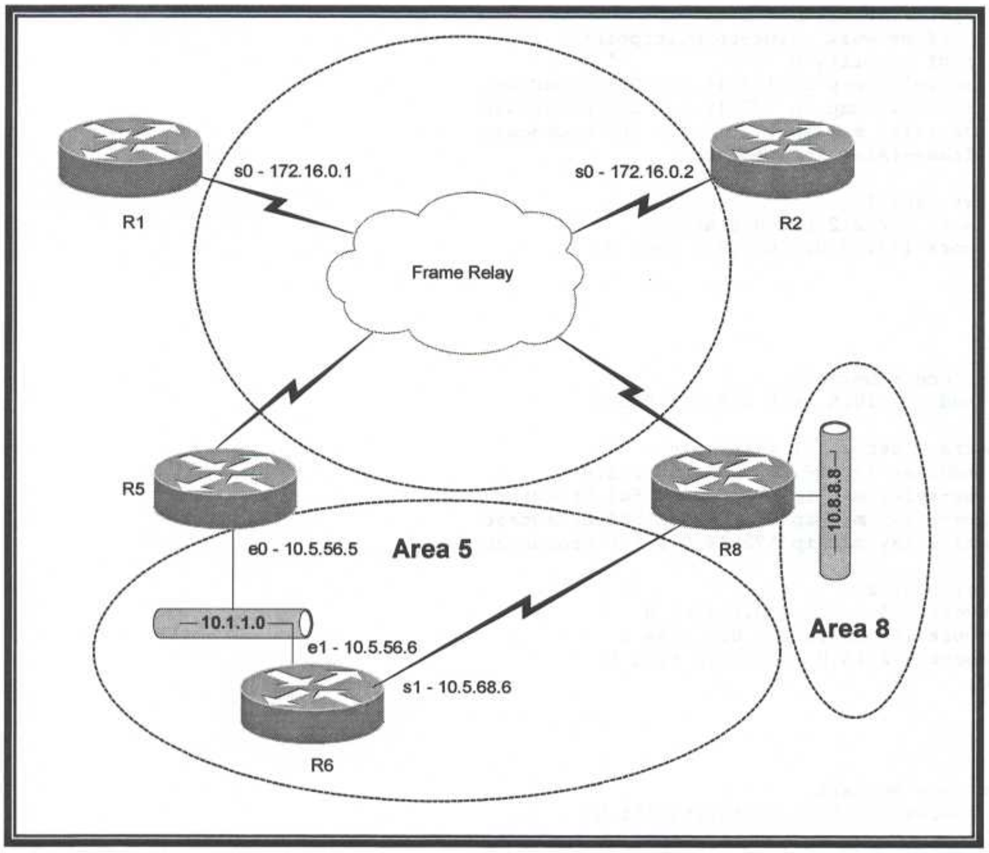
The Router ID can be manually configured beginning in IOS versions 12.0(1)T. If you have a requirement where you cannot change the loopback address, but must control the router ID this is the command you to use. The Router ID must be in IP address format.

```
r5(config)# router ospf 1
r5(config-router)# router-id 5.5.5.5
```

Virtual Links

For example, Area 5 has a connection to both Area 0 and Area 8. Area 8 normally is connected to Area 0 but its connection to Area 0 failed. The only way it can reach Area 0 is via Area 5. An administrator can configure Area 8 to use Area 5 as its transit area. This way it appears to OSPF as if Area 8 is directly connected to Area 0. Typically, virtual-links are used to solve a problem such as the one just mentioned, but are not considered good design practice.

Figure 7.3 *OSPF Virtual Link (prior to failure)*



OSPF is configured as shown in Figure 7.3. The pertinent configurations for each router are shown below. With this setup, there is no need for a virtual link since all ABR's have a connection to Area 0. Internal routers that only have a connection to a single area do not need to be connected to Area 0 provided at least one router in the area has a connection to Area 0. For example, R6 does not need to be connected to Area 0 since it is an internal router with interfaces in only a single area. R5 functions as the ABR for Area 5 to provide connectivity to Area 0 for the routers in Area 5.

ROUTER 1

```
interface Serial0
 ip address 172.16.0.1 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 ip ospf priority 0
 frame-relay map ip 172.16.0.5 105 broadcast
 frame-relay map ip 172.16.0.2 105 broadcast
 frame-relay map ip 172.16.0.8 105 broadcast
 no frame-relay inverse-arp
!
router ospf 1
 network 1.1.1.1 0.0.0.0 area 1
 network 172.16.0.1 0.0.0.0 area 0
```

Router 2

```
interface Serial0
 ip address 172.16.0.2 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 ip ospf priority 0
 frame-relay map ip 172.16.0.5 205 broadcast
 frame-relay map ip 172.16.0.1 205 broadcast
 frame-relay map ip 172.16.0.8 205 broadcast
 no frame-relay inverse-arp
!
router ospf 1
 network 2.2.2.2 0.0.0.0 area 2
 network 172.16.0.2 0.0.0.0 area 0
```

Router 5

```
interface Ethernet0
 ip address 10.5.56.5 255.255.255.0
!
interface Serial1.1 multipoint
 ip address 172.16.0.5 255.255.255.0
 frame-relay map ip 172.16.0.1 501 broadcast
 frame-relay map ip 172.16.0.2 502 broadcast
 frame-relay map ip 172.16.0.8 508 broadcast
!
router ospf 1
 network 5.5.5.5 0.0.0.0 area 5
 network 10.5.56.5 0.0.0.0 area 5
 network 172.16.0.5 0.0.0.0 area 0
```

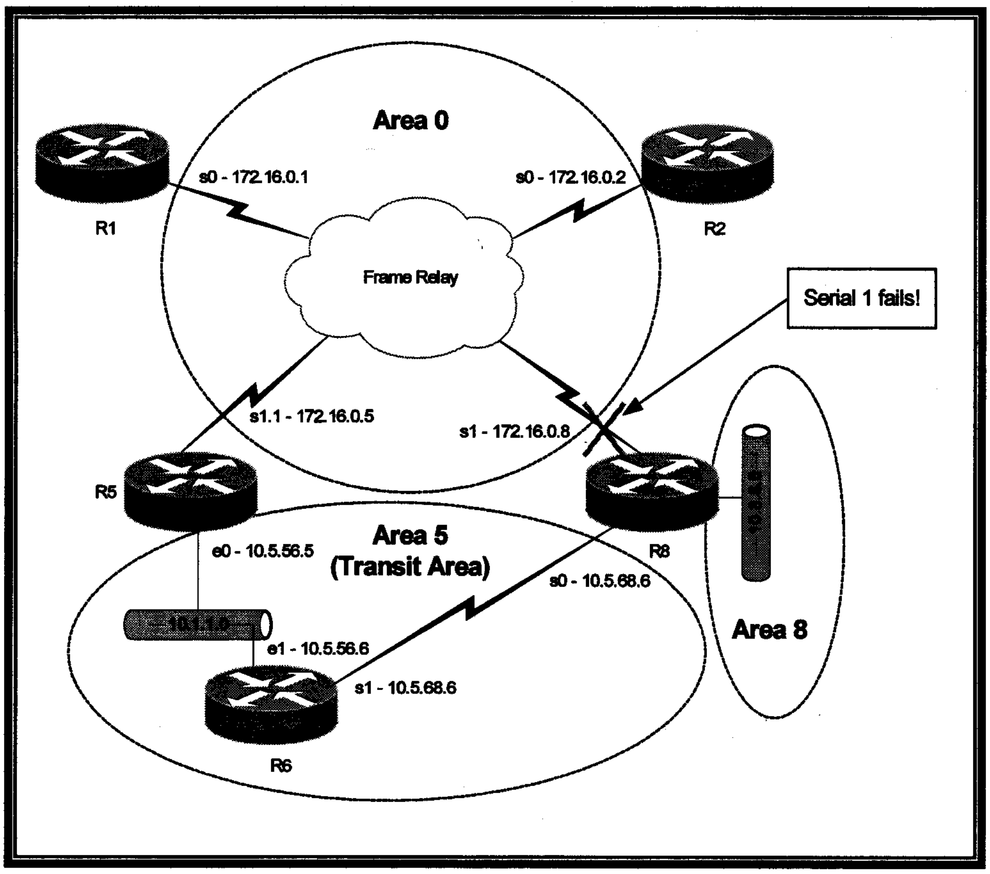
Router 6

```
interface Serial1
 ip address 10.5.68.6 255.255.255.0
!
interface Ethernet0
 ip address 10.5.56.6 255.255.255.0
!
router ospf 1
 network 6.6.6.6 0.0.0.0 area 5
 network 10.5.56.6 0.0.0.0 area 5
 network 10.5.68.6 0.0.0.0 area 5
```

Router 8

```
interface Serial0
 ip address 10.5.68.8 255.255.255.0
!
interface Serial1
 ip address 172.16.0.8 255.255.255.0
 encapsulation frame-relay
 ip ospf network point-to-multipoint
 ip ospf priority 0
 frame-relay map ip 172.16.0.5 805 broadcast
 frame-relay map ip 172.16.0.1 805 broadcast
 frame-relay map ip 172.16.0.2 805 broadcast
 no frame-relay inverse-arp
!
router ospf 1
 network 10.5.68.8 0.0.0.0 area 5
 network 10.8.8.8 0.0.0.0 area 8
```

Figure 7.4 *OSPF Virtual Link (after failure)*



If the frame-relay interface on R8 fails, the rest of the OSPF network will lose connectivity to Area 8. This is because that frame-relay interface was the only connectivity to Area 0. However, we can solve this problem by using Area 5 as a virtual-link (transit area) back to Area 0. R5 and R8 need to be configured for the virtual-link. R5 is chosen since it is the ABR between Area 5 and Area 0. R8 is chosen as the other end of the link because it is the ABR between Area 8 and Area 5.

The virtual-link configuration references the router ID and not an IP address of the directly connected network between R5 and R8.

```
r5(config)# router ospf 1
r5(config-router)# area 5 virtual-link 8.8.8.8

r8(config)# router ospf 1
r8(config-router)# area 5 virtual-link 5.5.5.5

r5# show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 8.8.8.8 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Ethernet0, Cost of using 10
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
    Adjacency State FULL (Hello suppressed)
```

```
r8# show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 5.5.5.5 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Ethernet0, Cost of using 10
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
    Adjacency State FULL (Hello suppressed)
    Index 1/3, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

OSPF Authentication

OSPF authentication provides some security by preventing an unauthorized router from injecting bogus routes on to the network. Without authentication, a user can connect an OSPF router or device to the LAN port in his or her cube and establish a neighbor relationship thereby giving them the ability to inject invalid routes on to the network.

There are two types of authentication: clear-text and MD5. Clear-text sends the password in the clear so anyone sniffing the segment can discover your OSPF password. MD5 hashes the password and makes it extremely difficult for unauthorized personnel to discover your password.

Configuring OSPF clear-text authentication

Configuring OSPF authentication requires two steps. The first step applies authentication to the interface and sets the actual password. The second step enables authentication for the area. Once authentication is enabled on the router, it will drop all of its neighbors until they are configured for authentication too.

Step 1 Apply authentication to the interface and set the password. This will need to be configured on all connected interfaces or neighbor relationships will not be formed.

```
r1(config)# interface serial0
r1(config-if)# ip ospf authentication-key cisco

r2(config)# interface serial0
r2(config-if)# ip ospf authentication-key cisco

r5(config)# interface serial1.1
r5(config-if)# ip ospf authentication-key cisco
```

Step 2 Configure the area to use authentication.

```
r1(config)# router ospf 1
r1(config-router)# area 0 authentication

r2(config)# router ospf 1
r2(config-router)# area 0 authentication

r5(config)# router ospf 1
r5(config-router)# area 0 authentication
```

You can verify that authentication is enabled for the interface using the **show ip ospf interface** command. You can also check which areas have authentication configured using the **show ip ospf** command. R1 and R2 have the same output so we did not display the show commands for R2.

```
r1# show ip ospf interface s0
Serial0 is up, line protocol is up
  Internet Address 172.16.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_MULTIPOINT, Cost: 64
```

```
   Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
   Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
     Hello due in 00:00:19
   Index 1/1, flood queue length 0
   Next 0x0(0)/0x0(0)
   Last flood scan length is 1, maximum is 1
   Last flood scan time is 0 msec, maximum is 0 msec
   Neighbor Count is 1, Adjacent neighbor count is 1
     Adjacent with neighbor 5.5.5.5
   Suppress hello for 0 neighbor(s)
   Simple password authentication enabled

 r1# show ip ospf
  Routing Process "ospf 1" with ID 1.1.1.1 and Domain ID 0.0.0.1
  Supports only single TOS(TOS0) routes
  Supports opaque LSA
  It is an area border router
  SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
  Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
  Number of external LSA 0. Checksum Sum 0x0
  Number of opaque AS LSA 0. Checksum Sum 0x0
  Number of DCbitless external and opaque AS LSA 0
  Number of DoNotAge external and opaque AS LSA 0
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  External flood list length 0
     Area BACKBONE(0)
         Number of interfaces in this area is 1
         Area has simple password authentication
         SPF algorithm executed 40 times
         Area ranges are
         Number of LSA 21. Checksum Sum 0x77685
         Number of opaque link LSA 0. Checksum Sum 0x0
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 12
         Flood list length 0
     Area 1
         Number of interfaces in this area is 1
         Area has no authentication
         SPF algorithm executed 2 times
         Area ranges are
         Number of LSA 9. Checksum Sum 0x377BD
         Number of opaque link LSA 0. Checksum Sum 0x0
         Number of DCbitless LSA 0
         Number of indication LSA 0
         Number of DoNotAge LSA 0
         Flood list length 0
```

For some reason, R5 does not show that the interface is configured for authentication. We believe this is probably because of this IOS version. This is why it is important to know more than one way to check a configuration.

OSPF Clear-Text Authentication Verification

```
r5# show ip ospf interface s1.1
Serial1.1 is up, line protocol is up
  Internet Address 172.16.0.5/24, Area 0
Process ID 1, Router ID 5.5.5.5, Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:04
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 1.1.1.1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)

r5# show ip ospf
 Routing Process "ospf 1" with ID 5.5.5.5
```

```
Supports only single TOS(TOS0) routes
It is an area border router
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
Number of external LSA 0. Checksum Sum 0x0
Number of DCbitless external LSA 0
Number of DoNotAge external LSA 0
Number of areas in this router is 2. 2 normal 0 stub 0 nssa
   Area BACKBONE(0)
       Number of interfaces in this area is 2
       Area has simple password authentication
       SPF algorithm executed 5 times
       Area ranges are
       Number of LSA 21. Checksum Sum 0x77685
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 12
   Area 5
       Number of interfaces in this area is 2
       Area has no authentication
       SPF algorithm executed 3 times
       Area ranges are
       Number of LSA 11. Checksum Sum 0x5E9A5
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
```

Configuring OSPF MD5 Authentication

Configuring OSPF authentication requires two steps. The first step applies authentication to the interface and sets the actual password. The second step enables authentication for the area. Once authentication is enabled on the router, it will drop all of its neighbors until they are configured for authentication too.

Step 1 Apply authentication to the interface and set the password. This will need to be configured on all connected interfaces or neighbor relationships will not be formed.

```
r1(config)# interface serial0
r1(config-if)# ip ospf message-digest-key 1 md5 0 cisco

r2(config)# interface serial0
r2(config-if)# ip ospf message-digest-key 1 md5 0 cisco

r5(config)# interface serial1.1
r5(config-if)# ip ospf message-digest-key 1 md5 0 cisco
```

Step 2 Configure the area to use authentication.

```
r1(config)# router ospf 1
r1(config-router)# area 0 authentication message-digest

r2(config)# router ospf 1
r2(config-router)# area 0 authentication message-digest
r5(config)# router ospf 1
r5(config-router)# area 0 authentication message-digest
```

OSPF MD5 Authentication Verification

```
r1# show ip ospf interface s0
Serial0 is up, line protocol is up
  Internet Address 172.16.0.1/24, Area 0
Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_MULTIPOINT, Cost:   64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:29
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 5.5.5.5
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1

r1# show ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1 and Domain ID 0.0.0.1
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 It is an area border router
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x0
 Number of opaque AS LSA 0. Checksum Sum 0x0
 Number of DCbitless external and opaque AS LSA 0
 Number of DoNotAge external and opaque AS LSA 0
 Number of areas in this router is 2. 2 normal 0 stub 0 nssa
 External flood list length 0
    Area BACKBONE(0)
        Number of interfaces in this area is 1
        Area has message digest authentication
        SPF algorithm executed 46 times
        Area ranges are
        Number of LSA 16. Checksum Sum 0x4E202
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 7
        Flood list length 0
    Area 1
        Number of interfaces in this area is 1
        Area has no authentication
        SPF algorithm executed 2 times
        Area ranges are
        Number of LSA 9. Checksum Sum 0x373BF
        Number of opaque link LSA 0. Checksum Sum 0x0
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
        Flood list length 0

r5# show ip ospf interface s1.1
Serial1.1 is up, line protocol is up
  Internet Address 172.16.0.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT,
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    Hello due in 00:00:10
  Neighbor Count is 2, Adjacent neighbor count is 2
    Adjacent with neighbor 1.1.1.1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1

r5# show ip ospf
 Routing Process "ospf 1" with ID 5.5.5.5
 Supports only single TOS(TOS0) routes
 It is an area border router
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Minimum LSA interval 5 secs. Minimum LSA arrival 1 secs
 Number of external LSA 0. Checksum Sum 0x0
 Number of DCbitless external LSA 0
 Number of DoNotAge external LSA 0
 Number of areas in this router is 2. 2 normal 0 stub 0 nssa
    Area BACKBONE(0)
        Number of interfaces in this area is 2
        Area has message digest authentication
        SPF algorithm executed 11 times
        Area ranges are
        Number of LSA 27. Checksum Sum 0x9344D
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 14
    Area 5
        Number of interfaces in this area is 2
        Area has no authentication
        SPF algorithm executed 8 times
        Area ranges are
        Number of LSA 14. Checksum Sum 0x7D390
        Number of DCbitless LSA 0
        Number of indication LSA 0
        Number of DoNotAge LSA 0
```

Configuring OSPF Virtual-Link Clear-Text Authentication

If Area 0 is configured for authentication, then the virtual link must also be configured for authentication. R5 is already configured for Area 0 authentication. This configuration assumes that Area 0 is already properly configured for clear-text authentication including the **area 0 authentication** command.

Step 1 Configure the router that has a working connection to Area 0 for the virtual link and authentication key.

```
r5(config)# router ospf 1
r5(config-router)# area 5 virtual-link 8.8.8.8 authentication-key cisco
```

Step 2 Configure the router that does not have a working connection to Area 0 for the virtual link and authentication key. Additionally, configure **area 0 authentication** even though this router does not have a physical interface connected to Area 0.

```
r8(config)# router ospf 1
r8(config-router)# area 0 authentication
r8(config-router)# area 5 virtual-link 5.5.5.5 authentication-key cisco
```

OSPF Virtual-Link Clear-Text Authentication Verification

```
r5# show ip ospf virtual-links
Virtual Link OSPF_VL3 to router 8.8.8.8 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Ethernet0, Cost of using 74
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:04
    Adjacency State FULL (Hello suppressed)

r8# show ip ospf virtual-links
Virtual Link OSPF_VL1 to router 5.5.5.5 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Serial0, Cost of using 74
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
    Adjacency State FULL (Hello suppressed)
    Index 1/2, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
  Simple password authentication enabled
```

CONFIGURING OSPF VIRTUAL-LINK MD5 AUTHENTICATION

If Area 0 is configured for MD5 authentication, then the virtual link must also be configured for MD5 authentication. R5 is already configured for Area 0 MD5 authentication. This configuration assumes that Area 0 is already properly configured for MD5 authentication including the **area 0 authentication message-digest** command.

Step 1 Configure the router that has a working connection to Area 0 for the virtual link and authentication key.

```
r5(config)# router ospf 1
r5(config-router)# area 5 virtual-link 8.8.8.8 message-digest-key 1 md5 cisco
```

Step 2 Configure the router that does not have a working connection to Area 0 for the virtual link and authentication key. Additionally, configure **area 0 authentication** even though this router does not have a physical interface connected to Area 0.

```
r8(config)# router ospf 1
r8(config-router)# area 5 virtual-link 5.5.5.5 message-digest-key 1 md5 cisco
r8(config-router)# area 0 authentication message-digest
```

OSPF VIRTUAL-LINK MD5 VERIFICATION

```
r5# show ip ospf virtual-links
Virtual Link OSPF_VL2 to router 8.8.8.8 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Ethernet0, Cost of using 74
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
    Adjacency State FULL (Hello suppressed)
  Message digest authentication enabled
    Youngest key id is 1

r8# show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 5.5.5.5 is up
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 5, via interface Serial0, Cost of using 74
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
    Adjacency State FULL (Hello suppressed)
    Index 1/2, retransmission queue length 0, number of retransmission 1
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
  Message digest authentication enabled
    Youngest key id is 1
```

TYPICAL GOTCHAS

- Mismatched OPSF network types over Frame Relay
- Forgetting neighbor statements in a NON-BROADCAST network
- Remembering to configure the spokes to a priority of 0 for a Hub and Spoke Frame Relay that is utilizing a DR
- Missing virtual links (All areas must touch area 0)
- Losing virtual-link connections due to changing router ID's – statically set the OSPF router ID
- Not configuring the virtual link with the same authentication as area 0 – don't forget the far end
- Adding extra spaces at the end of passwords – use 'debug ip ospf adj'
- Mismatched authentication keys or passwords

CHAPTER 8

BORDER GATEWAY PROTOCOL (BGP)

BGP is used to route between Autonomous Systems and is the routing protocol for the Internet. Configuration of BGP can be quite complicated and there are many options. We will try to cover most of the BGP topics that may be on the lab exam.

BGP Peers

BGP requires that routers establish a peer relationship. Unlike OSPF, this neighbor (peer) relationship must be manually configured. Routers are considered peers or neighbors whenever they open up a TCP session to exchange routing information. When routers communicate for the first time, they exchange their entire routing table. From then on, they send only incremental updates. BGP uses TCP as its transport protocol, via port 179.

Internal BGP (IBGP)

- Exchanges routing information within the same AS between routers.
- IBGP routers must be fully meshed
- All IBGP routers must have the same BGP routing table (only EBGP links can adjust or filter BGP routes)

External BGP (EBGP)

- Used when routers belong to different AS's and exchange BGP updates.
- BGP must be synchronized with the IGP (IGP's include such routing protocols as OSPF, RIP, EIGRP, etc.) if the AS provides transit service for other AS's. Synchronization helps prevent BGP from advertising an internal route that is no longer available via the IGP.
- When to disable synchronization:
  - Your AS does not transfer traffic from one AS to another (transit AS)
  - All the transit routers on your AS are running BGP

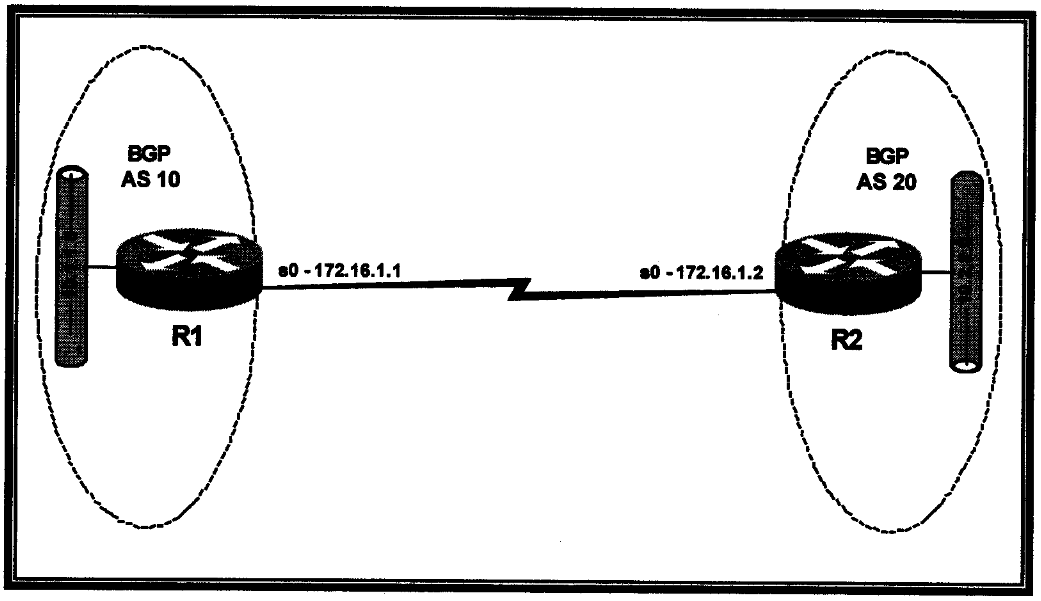
Basic BGP Configuration

Enable BGP using a local BGP AS number assigned by InterNIC (for a production environment). During the lab exam, you will use AS numbers assigned by the exam instructions.

There are a few rules when configuring BGP. Neighbors must be configured on both sides. Also, neighbors must be directly connected or have a specific route (a default route will not work) to the neighbor. Multihop must be configured if the neighbors are not directly connected. Networks configured must have a match in the routing table in order for BGP to advertise the route.

To configure BGP, first start the BGP routing process. Then advertise networks in to BGP (if applicable). Finally, configure your BGP peers.

Figure 8.1 *Basic BGP topology*



```
r1(config)# router bgp 10

r2(config)# router bgp 20
```

Configure the networks you want to advertise.

```
r1(config-router)# network 10.1.1.0 mask 255.255.255.0

r2(config-router)# network 10.2.2.0 mask 255.255.255.0
```

Specify BGP neighbors and IP address.

```
r1(config-router)# neighbor 172.16.1.2 remote-as 20

r2(config-router)# neighbor 172.16.1.1 remote-as 10
```

Note Once you have configured basic BGP, you'll typically need to clear the BGP session for any new changes to take effect by entering the `clear ip bgp *` command.

BGP Neighbor Verification

Once neighbors are configured, verify that you have a valid TCP and BGP connection.

```
r1# show ip bgp neighbors
BGP neighbor is 172.16.1.2,  remote AS 20, external link
  BGP version 4, remote router ID 10.2.2.22
  BGP state = Established, up for 00:01:20
  Last read 00:00:19, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(new)
    Address family IPv4 Unicast: advertised and received
  Received 11 messages, 0 notifications, 0 in queue
  Sent 8 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 3, neighbor version 3
  Index 1, Offset 0, Mask 0x2
  1 accepted prefixes consume 36 bytes
  Prefix advertised 1, suppressed 0, withdrawn 0
  Number of NLRIs in the update sent: max 1, min 0

  Connections established 2; dropped 1
  Last reset 00:01:59, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 172.16.1.1, Local port: 11000
Foreign host: 172.16.1.2, Foreign port: 179

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x3912A454):
Timer          Starts    Wakeups            Next
Retrans             6          0             0x0
TimeWait            0          0             0x0
AckHold             6          2             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss:    5543254  snduna:    5543409  sndnxt:    5543409     sndwnd:  16230
irs: 3704107542  rcvnxt: 3704107716  rcvwnd:      16211  delrcvwnd:    173

SRTT: 165 ms, RTTO: 1172 ms, RTV: 1007 ms, KRTT: 0 ms
minRTT: 24 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, nagle

Datagrams (max data segment is 1460 bytes):
Rcvd: 8 (out of order: 0), with data: 6, total data bytes: 173
Sent: 9 (retransmit: 0), with data: 5, total data bytes: 154
```

To verify the routes being advertised by BGP, use the **show ip bgp** command.

```
r1# show ip bgp
BGP table version is 3, local router ID is 10.1.1.1
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
*> 10.1.1.0/24      0.0.0.0                  0         32768 i
*> 10.2.2.0/24      172.16.1.2               0             0 20 i
```

SYNCHRONIZATION

BGP must be synchronized with the IGP (Interior Gateway Protocol, such as OSPF or EIGRP) unless the administrator specifically disables it. With synchronization enabled, BGP waits until the IGP has propagated routing information across the autonomous system before advertising routes to other AS's.

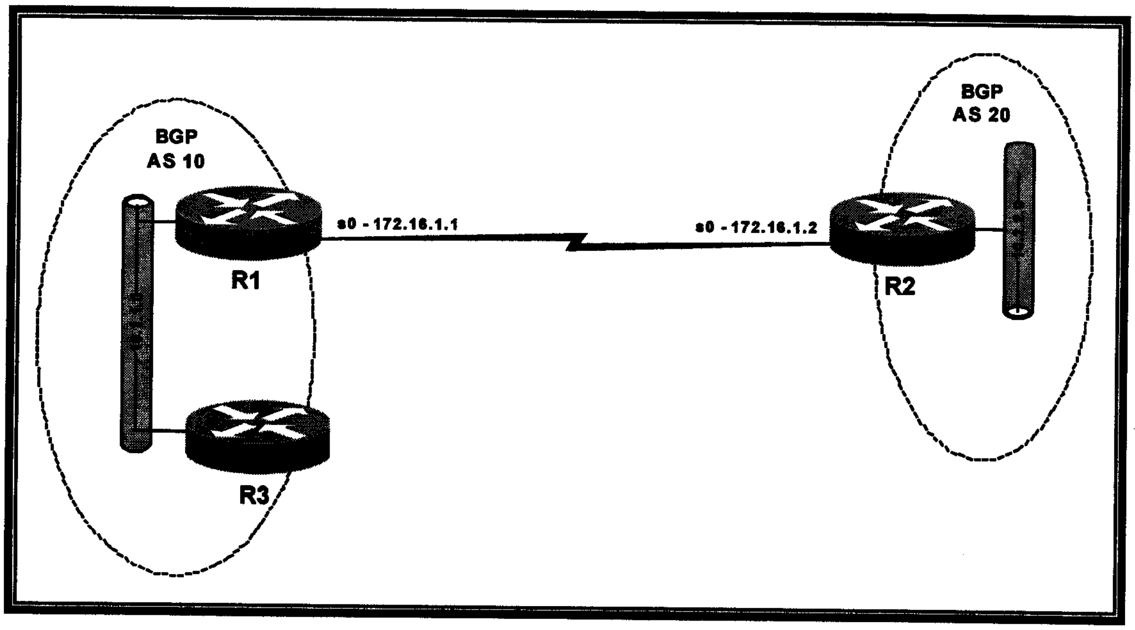
It is always recommended to disable synchronization with the BGP `no synchronization` command unless instructed otherwise by the exam instructions.

```
r1(config)# router bgp 10
r1(config-router)# no synchronization
```

NEXT-HOP-SELF

An EBGP learned route cannot be installed in the routing table of IBGP connected routers unless the route's next-hop address is reachable. The **show ip bgp** command will show all networks advertise via BGP. However, these networks may not be installed in the routing table. For example, since R3 does not have a route to 172.16.1.2, it cannot put the 10.2.2.0 network in its routing table. Remember the rules of BGP state that in order for a route to be valid it must know how to reach the next hop.

Figure 8.2 *BGP Next-Hop-Self*



```
r3# show ip bgp
BGP table version is 2, local router ID is 10.2.2.3
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
```

```
*>i10.1.1.0/24      10.1.1.1                 0    100      0 i
* i10.2.2.0/24      172.16.1.2               0    100      0 20 i

r3# show ip route 172.16.1.2
% Network not in table
```

The problem above should be clear. R2 advertises the 10.2.2.0 network to R1 with a next hop of 172.16.1.2. R1 passes this same exact advertisement to R3. R3 does not have a route to the next hop. We can fix this problem by either adding the 172.16.1.0 network to the IGP (or use a static route) or we can use the `next-hop-self` command. Next-hop-self changes the next hop to its own router.

```
r1(config)# router bgp 10
r1(config-router)# neighbor 10.1.1.3 next-hop-self

r3# show ip bgp
BGP table version is 4, local router ID is 10.2.2.3
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
*>i10.1.1.0/24      10.1.1.1                 0    100      0 i
*>i10.2.2.0/24      10.1.1.1                 0    100      0 20 i

r3# ping 10.2.2.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/44/88 ms
```

Note Try the above configuration without disabling synchronization. If synchronization is enabled on R3, it will not put the 10.2.2.0 network in the routing table. This is because it expects to see this route advertised via the IGP. It will show up on R2 because synchronization does not apply for routes learned via EBGP.

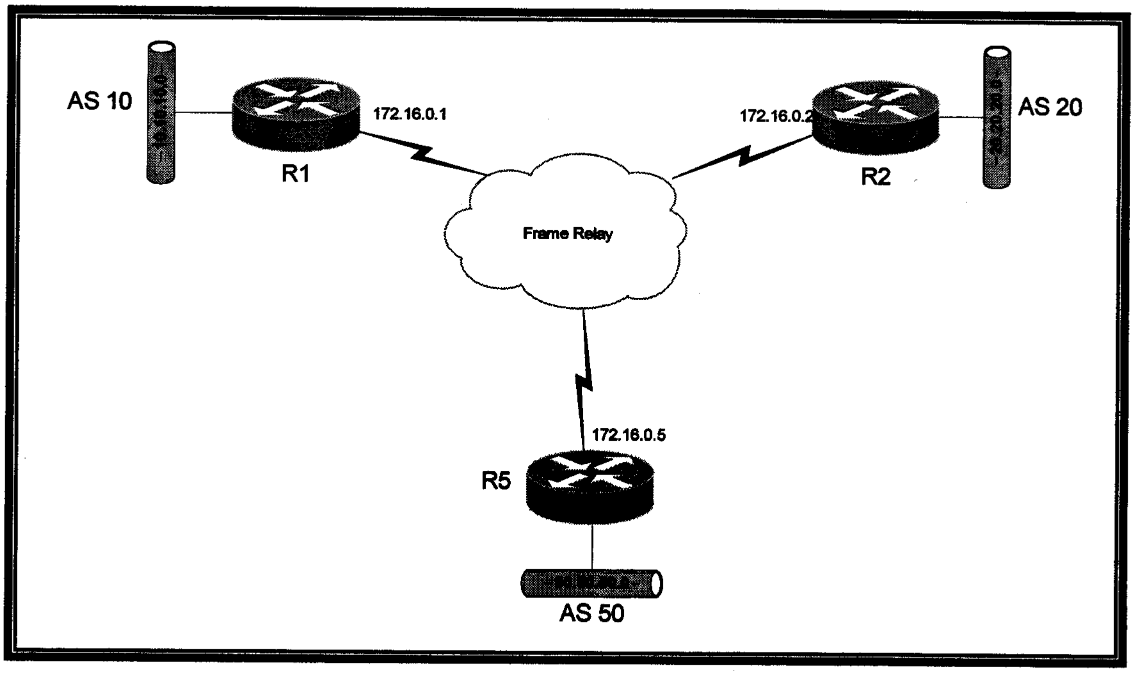
TRANSIT AS

If an AS has two or more connections to the Internet (or other AS's that have connectivity to each other), by default, traffic not destined for your AS may pass through your routers if your AS is the best path to a destination. You can prevent this by using one of three ways.

1. Use filters and only allow your network(s) to be the source of traffic
2. Use filters on your neighbors in conjunction with the `ip as-path access-list 1 permit ^$` command
3. Use communities and the no-export tag

The following three examples illustrate how to prevent your AS from becoming a transit AS. Refer to figure 8.3 for the network layout and current configuration.

Figure 8.3 *BGP Transit AS*



```
hostname r1
!
interface Ethernet0
 ip address 10.10.10.10 255.255.255.0
!
interface Serial0
 ip address 172.16.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.0.5 105 broadcast
 frame-relay map ip 172.16.0.2 105 broadcast
 no frame-relay inverse-arp
!
router bgp 10
 neighbor 172.16.0.5 remote-as 50
 network 10.10.10.0 mask 255.255.255.0

hostname r2
interface Ethernet0
 ip address 20.20.20.20 255.255.255.0
!
interface Serial0
 ip address 172.16.0.2 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.0.5 205 broadcast
 frame-relay map ip 172.16.0.1 205 broadcast
 no frame-relay inverse-arp
!
router bgp 20
 neighbor 172.16.0.5 remote-as 50
 network 20.20.20.0 mask 255.255.255.0

hostname r5
interface Ethernet0
 ip address 50.50.50.50 255.255.255.0
 no ip directed-broadcast
```

```
!
interface Serial1
 no ip address
 encapsulation frame-relay
 no frame-relay inverse-arp
!
interface Serial1.1 multipoint
 ip address 172.16.0.5 255.255.255.0
 frame-relay map ip 172.16.0.1 501 broadcast
 frame-relay map ip 172.16.0.2 502 broadcast
!
router bgp 50
 neighbor 172.16.0.1 remote-as 10
 neighbor 172.16.0.2 remote-as 20
 network 50.50.50.0 mask 255.255.255.0
```

EXAMPLE 1 – BASIC FILTERS

If you were the administrator for AS 50 you would probably not want traffic from AS 10 going through your AS to reach AS 20 (assuming these were regular Internet connections). For the purpose of these examples, we are going to assume this is undesired.

Prior to configuring your filters, you need to identify which networks AS 50 should be allowed to advertise. Our lab is quite simple as we only advertise 50.50.50.0. Therefore, we configure a prefix-list to only allow that network. The second prefix-list is just for demonstration purposes since, like an access-list, there is an explicit deny at the end of a prefix-list. Note that prefix-lists can use names, such as AS50, or numbers. The "seq" is a sequence number. This allows you to add or remove lines to the prefix-list without removing the entire list. But, you must leave a few available numbers between each line. We like to use 5 numbers between each prefix-list line. Regular access-lists can be difficult to work with and make changes because of this limitation. Make sure you understand all your options with prefix-lists including "le" (less than) and "ge" (greater than).

```
r5(config)# ip prefix-list AS50 seq 5 permit 50.50.50.0/24
r5(config)# ip prefix-list AS50 seq 10 deny 0.0.0.0/0 le 32
```

Unlike a regular access-list, a prefix-list can be applied directly to a neighbor. We start by adding this prefix-list to one neighbor. After clearing the BGP session, we should notice that R2 no longer sees R1's 10.10.10.0 network. If R1 does not receive a route from AS 50 it cannot use AS 50 to route to AS 10.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.2 prefix-list AS50 out
```

Notice that the 10.10.10.0 network is not received as a BGP route.

```
r2# show ip bgp
BGP table version is 9, local router ID is 2.2.2.2
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
*> 20.20.20.0/24    0.0.0.0                  0         32768 i
*> 50.50.50.0/24    172.16.0.5               0             0 50 i
```

Example 2 – Filters with ip as-path command

Example 1 is simple to configure and easy to understand, but what if we have several hundred routes and most of them cannot be aggregated? Let's also assume we have dozens of Internet routers running BGP. Clearly, this is not scalable so we have to figure out a better way.
Using filters with ip as-path we can filter based on the AS Path. Since we want to prevent our AS from becoming a transit AS, we only want to allow traffic that originated from our AS to communicate out to the Internet. So, we configure an **ip as-path access-list** and only allow our AS. Remember that traffic that originates from an AS does not get an AS tag until it reaches another AS. When the traffic leaves our AS, it will have a blank AS Path. This means we need to only permit traffic with no AS Path! The tricky part of this type of filtering is understanding the regular expression ^$. The ^ notates the beginning of the AS Path and the $ notates the end of the path. You should notice it matches on a blank path, which is what we want.

```
r5(config)# router bgp 50
r5(config-router)# ip as-path access-list 1 permit ^$
```

Once we configure the ip as-path we have to apply it to a neighbor. Remember that these filters are applied to outbound updates.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.2 filter-list 1 out
```

Example 3 – Communities

Communities allows you to dynamically control how your EBGP peers handle routes you send to them. There are two examples involving communities. The first example illustrates how the administrator of AS 50, the transit AS, could be configured to tag all incoming routes with the no-export community. The second example is how AS 10 can prevent AS 50 from being a transit AS. It is important to know both ways in case the lab exam does not allow you to make changes on AS 50 to accomplish the task.

Communities Inbound

Step 1 Configure the access-list and route-map. The access-list is used by the route-map so it knows which routes to tag for communities.

```
r5(config)# access-list 1 permit 10.10.10.0 0.0.0.255
r5(config)# access-list 1 permit 20.20.20.0 0.0.0.255
```

Step 2 Configure the route-map and match the access-list just created. Set these routes to have the community attribute of "no-export."

```
r5(config)# route-map NO_EXPORT permit 10
r5(config-route-map)# match ip address 1
r5(config-route-map)# set community no-export
```

Step 3 Apply that route-map to the neighbor.

```
r5(config-router)# neighbor 172.16.0.1 route-map NO_EXPORT in
r5(config-router)# neighbor 172.16.0.2 route-map NO_EXPORT in
```

Step 4 Verify that the routes received by the peer have the no-export tag.

```
r5# show ip bgp 10.10.10.0
BGP routing table entry for 10.10.10.0/24, version 6
```

```
Paths: (1 available, best #1, not advertised to EBGP peer)
  Not advertised to any peer
  10
    172.16.0.1 from 172.16.0.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best, ref 2
      Community: no-export
```

COMMUNITIES OUTBOUND

Let's assume that we are now the administrator for AS 10 and AS 20. We can prevent AS t0 from being a transit AS using communities. We simply tell AS 50 to not advertise (or export) our networks to any other external AS.

The main difference between tagging outbound versus inbound is that the neighbor must be configured to send the community property to its neighbors. When tagging incoming routes there is no such requirement.

Step 1 Configure the access-lists and route-maps.

```
r1(config)# access-list 1 permit 10.10.10.0 0.0.0.255

r2(config)# access-list 1 permit 20.20.20.0 0.0.0.255
```

Step 2 Configure the route-map and match the access-list just created.

```
r1(config)# route-map SEND_COMMUNITY permit 10
r1(config-route-map)# match ip address 1
r1(config-route-map)# set community no-export

r2(config)# route-map SEND_COMMUNITY permit 10
r2(config-route-map)# match ip address 1
r2(config-route-map)# set community no-export
```

Step 3 Apply that route-map to the neighbor.

```
r1(config-router)# neighbor 172.16.0.5 route-map SEND_COMMUNITY out

r2(config-router)# neighbor 172.16.0.5 route-map SEND_COMMUNITY out
```

Step 4 Enable the router to send the community attributes to its neighbor.

```
r1(config-router)# neighbor 172.16.0.5 send-community

r2(config-router)# neighbor 172.16.0.5 send-community
```

Step 5 Verify that the routes received by the peer have the no-export tag.

```
r5# show ip bgp 10.10.10.0
BGP routing table entry for 10.10.10.0/24, version 6
Paths: (1 available, best #1, not advertised to EBGP peer)
  Not advertised to any peer
  10
    172.16.0.1 from 172.16.0.1 (1.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, external, best, ref 2
      Community: no-export
```

MD5 Authentication

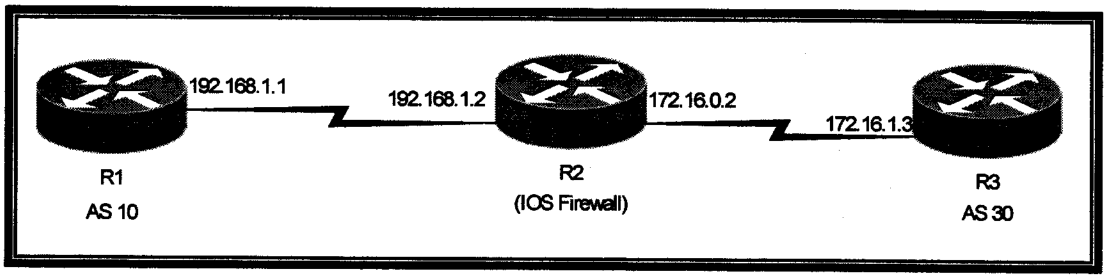
MD5 passwords is the only supported authentication type. There is no clear text authentication option for BGP.

```
r1(config)# router bgp 10
r1(config-router)# neighbor 172.16.1.2 password cisco
```

EBGP Multihop

BGP normally requires that neighbors be on directly connected networks. If they are not, they can still become neighbors. However, you need to permit this relationship and limit the number of hops they can be away. If you do not specify the number of hops, it will default to 255. This situation can be useful if another router or firewall separates two BGP peers as shown in Figure 8.4.

Figure 8.4 *BGP Multihop*



```
r1(config)# router bgp 10
r1(config-router)# neighbor 172.16.1.3 ebgp-multihop 2

r3(config)# router bgp 30
r3(config-router)# neighbor 192.168.1.1 ebgp-multihop 2
```

BGP Path Selection

BGP will select one path as the best path to reach a destination. This path is put into the BGP routing table and then propagated to its neighbors. The criteria for selecting the path for a destination are as follows.

1. If a path has a next hop that is inaccessible, the update is not considered
2. Weight (largest is preferred)
3. Local preference (largest is preferred)
4. AS-path (shortest is preferred)
5. Origin path (lowest is preferred)
6. MED (lowest is preferred)
7. External path is preferred to an internal path
8. Path through the closest IGP neighbor

9. Lowest IP address as specified by the BGP router ID. The router ID will be the router's loopback address is one is present.

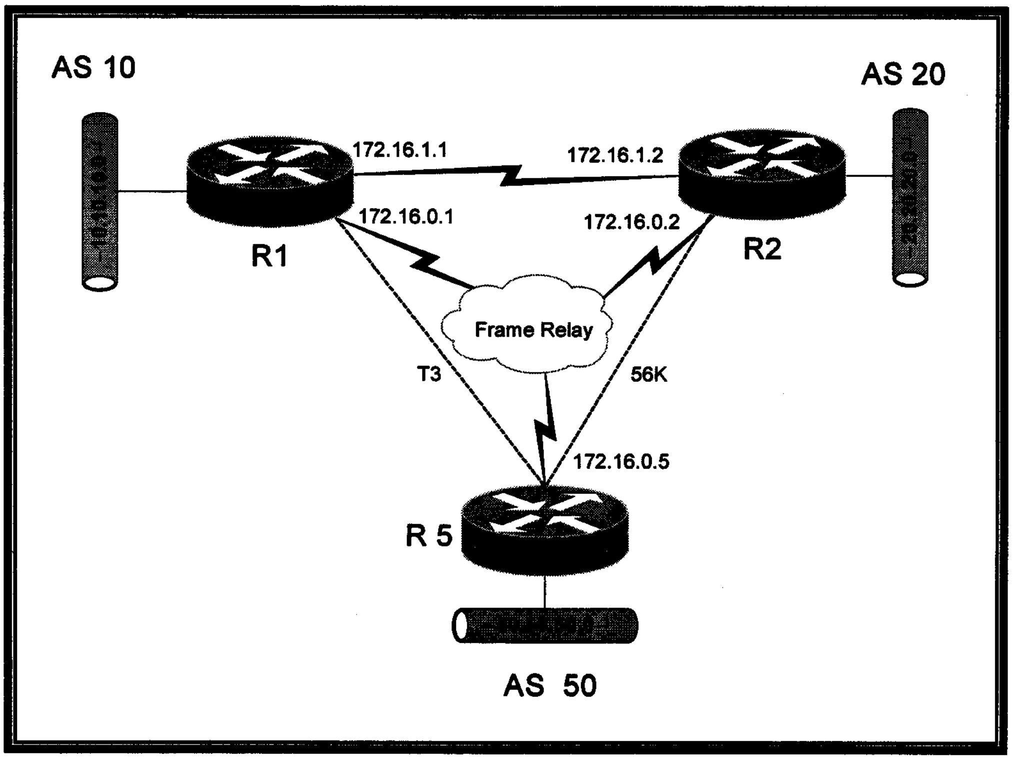
In the following sections, we are going to explain how to manipulate path selection based on the most common configuration options.

WEIGHT

Weight is Cisco proprietary and not a well-known BGP attribute. It is also local to the router which means this attribute is not passed to ANY other router.

In order to demonstrate path selection, we've made a slight modification to our previous network layout. We've connected AS 10 and AS 20 directly to each other. Now, all the AS's have direct connection to each other as shown in Figure 8.5.

Figure 8.5 *BGP path determination layout*



To check the current weight of the networks received, enter the **show ip bgp** command. Notice that AS 50 is seeing two paths to reach each EBGP route. It will select a path based on the shortest path since the weight, local preference, and metric are all equal. The default weight for local networks is 32768.

```
r5# show ip bgp
BGP table version is 5, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
```

```
*> 10.10.10.0/24     172.16.0.1               0              0 10 i
*                    172.16.0.2                              0 20 10 i
*  20.20.20.0/24     172.16.0.1                              0 10 20 i
*>                   172.16.0.2               0              0 20 i
*> 50.50.50.0/24     0.0.0.0                  0          32768 i
```

According to Figure 8.5, our connection to from AS 50 to AS 10 is a T3 and the connection to AS 20 is a very slow 56k link. We want all of the routes received from AS 10 to be preferred even when trying to reach the networks that belong to AS 20. So, we adjust the weight of all routes received from AS 10 (neighbor 172.16.0.1) so it has a higher weight and therefore preferred. To verify this path just enter the **show ip route** command. Without the weight command, we would see that the preferred route to the 20.20.20.0 network would be via the 172.16.0.2 neighbor since it has the shortest path. However, once we modify the weight, all networks received from 172.16.0.1 should be preferred as shown in our example below.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 weight 1024

r5# show ip bgp
BGP table version is 6, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24    172.16.0.1               0          1024 10 i
*                   172.16.0.2                             0 20 10 i
*> 20.20.20.0/24    172.16.0.1                          1024 10 20 i
*                   172.16.0.2               0             0 20 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [20/0] via 172.16.0.1, 00:01:19
     20.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0/24 is directly connected, Serial1.1
C       172.16.65.0/24 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
B       10.10.10.0 [20/0] via 172.16.0.1, 00:01:22
```

Weight can also be set using a route-map. What if we wanted to set the weight on specific routes instead of all routes from a neighbor? We could configure a route-map that sets the weight for certain routes and does not set the weight for other routes. Using the example above, we decide that we want the weight for all routes from AS 10 to remain 1024 except for the 20.20.20.0 route. We discovered that although our T3 is very fast to AS 10, it is still much faster to use the 56k link and go directly to AS 20 to reach the 20.20.20.0 route.

Step 1 Configure access-lists that will be used to match traffic. In our example, we want to match the network from AS 20.

```
r5(config)# access-list 10 permit 20.20.20.0 0.0.0.255
```

```
r5(config)# access-list 11 permit any
```

Step 2 Configure the first sequence of the route-map, which will assign a weight of 0 to the route(s) that match access-list 10.

```
r5(config)# route-map SETWEIGHT permit 10
r5(config-route-map)# match ip address 10
r5(config-route-map)# set weight 0
```

Step 3 Configure the last sequence number, which will assign all other routes

```
r5(config)# route-map SETWEIGHT permit 20
r5(config-route-map)# match ip address 11
r5(config-route-map)# set weight 1024
```

Step 4 Apply the new route-map to the neighbor. This route-map is applied to incoming updates.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 route-map SETWEIGHT in
```

BGP Weight Verification

After we clear the BGP session, we check our `show ip bgp` and `show ip route` commands to make sure we achieved the desired results. You should notice the ">" next to each network. If a network is received from multiple sources, only one will be preferred. Because we set a higher weight for the 10.10.10.0 network from 172.16.0.1, it should be considered "best."

```
r5# show ip bgp
BGP table version is 5, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24    172.16.0.1               0          1024 10 i
*                   172.16.0.2                             0 20 10 i
*  20.20.20.0/24    172.16.0.1                             0 10 20 i
*>                  172.16.0.2               0             0 20 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [20/0] via 172.16.0.2, 00:01:02
     20.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0/24 is directly connected, Serial1.1
C       172.16.65.0/24 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
B       10.10.10.0 [20/0] via 172.16.0.1, 00:00:53
```

LOCAL PREFERENCE

Local preference is at the highest level of the BGP decision process after weight. Like weight, the larger the number the more preferred the path. Try to remember this since it is opposite of metric or administrative distance which prefer a lower number.

The configuration of local preference is almost identical to the configuration of route-maps with the weight option. Instead, we set the local-preference instead of weight. In our example, we prefer the path through AS 10 to reach all networks except the 20.20.20.0 network. Remember that route-maps are processed in sequence number order. Notice that a local-preference for the 20.20.20.0 network is not set. We are leaving it at the default which is no value assigned. It is essentially a local preference of zero. Note that the value of 200 is a number we just made up. The local preference can be any number from 0 to 4294967295.

```
r5(config)# access-list 10 permit 20.20.20.0 0.0.0.255
r5(config)# access-list 11 permit any

r5(config)# route-map SETLOCALPREF permit 10
r5(config-route-map)# match ip address 10

r5(config)# route-map SETLOCALPREF permit 20
r5(config-route-map)# match ip address 11
r5(config-route-map)# set local-preference 200

r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 route-map SETLOCALPREF in
```

The 10.10.10.0 network is the only network received from AS 10 that will match the second sequence number. So, it will be the only network that has a local preference. If AS 10 advertised more than one network, (except for 20.20.20.0 since it is specifically excluded from the sequence that assigns the local preference) they would also be assigned a local preference of 200.

```
r5# show ip bgp
BGP table version is 5, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24    172.16.0.1               0    200      0 10 i
*                   172.16.0.2                             0 20 10 i
*  20.20.20.0/24    172.16.0.1                             0 10 20 i
*>                  172.16.0.2               0             0 20 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [20/0] via 172.16.0.2, 00:02:42
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C       172.16.0.0/24 is directly connected, Serial1.1
C       172.16.65.0/24 is directly connected, Serial1.2
```

```
     10.0.0.0/24 is subnetted, 1 subnets
B       10.10.10.0 [20/0] via 172.16.0.1, 00:02:31
```

AS Path Manipulation

After weight and local preference, BGP selects its route based on the shortest AS path. The AS path can be viewed with the **show ip bgp** command. You should notice that the ">" is next to the next hop that has the shortest path. For example, the shortest path to 20.20.20.0 is via 172.16.0.2.

```
r5# show ip bgp
BGP table version is 5, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  10.10.10.0/24    172.16.0.2                             0 20 10 i
*>                  172.16.0.1               0             0 10 i
*> 20.20.20.0/24    172.16.0.2               0             0 20 i
*                   172.16.0.1                             0 10 20 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 I
```

What if you had to prefer the path via AS 10 for all networks and you could not use weight or local preference? Manipulate the AS path via R2 to make it longer and therefore less preferable. In this next example, we are going to prefer the path to 20.20.20.0 via AS 1. To do this we are going to increase the AS Path length for the 20.20.20.0 network from R2. This is accomplished using a route-map and prepending AS paths.

```
r5(config)# access-list 11 permit any

r5(config)# route-map SETAS permit 10
r5(config-route-map)# match ip address 11
r5(config-route-map)# set as-path prepend 20 20

r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.2 route-map SETAS in
```

TIP Only use your own AS number when using the **set as-path prepend** command. Using other AS numbers may cause routing problems.

AS Path Manipulation Verification

As you can see in the **show ip bgp** command below, the path through 172.16.0.1 is preferred over the path through 172.16.0.2 because the AS path length is longer.

```
r5# show ip bgp
BGP table version is 6, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24    172.16.0.1               0             0 10 i
*                   172.16.0.2                             0 20 20 20 10 i
*> 20.20.20.0/24    172.16.0.1                             0 10 20 i
*                   172.16.0.2               0             0 20 20 20 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i


r5# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [20/0] via 172.16.0.1, 00:03:02
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
B       10.10.10.0 [20/0] via 172.16.0.1, 00:03:02
```

METRIC AND MED'S

Metrics can be used to influence path determination for an AS. Metrics can also be sent outside the AS to influence another AS's path back to it's own AS. Influencing another AS's metrics is called a MED.

METRIC CONFIGURATION

Step 1 Configure the access-list to have the metrics applied. Since we are going to apply metrics to all incoming routes, we do not need to be very granular. A simple "permit any" is sufficient for now.

```
r5(config)# access-list 11 permit any
```

Step 2 Configure the route-maps to match networks with the desired metric. SETMETRIC1 will be applied to updates received from AS 10. SETMETRIC2 will be applied to updates received from AS 20.

```
r5(config)# route-map SETMETRIC1 permit 10
r5(config-route-map)# match ip address 11
r5(config-route-map)# set metric 100

r5(config)# route-map SETMETRIC2 permit 10
r5(config-route-map)# match ip address 11
r5(config-route-map)# set metric 200
```

Step 3 Apply the route-maps to the neighbors and clear BGP.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 route-map SETMETRIC1 in
r5(config-router)# neighbor 172.16.0.2 route-map SETMETRIC2 in
```

METRIC VERIFICATION

```
r5# show ip bgp
BGP table version is 4, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.10.0/24    172.16.0.1             100             0 10 i
*                   172.16.0.2             200             0 20 10 i
*  20.20.20.0/24    172.16.0.1             100             0 10 20 i
*>                  172.16.0.2             200             0 20 i
```

```
Notice the metrics in our show ip route.
```

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [20/200] via 172.16.0.2, 00:01:52
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.0.0 is directly connected, Serial1.1
C       172.16.65.0 is directly connected, Serial1.2
     10.0.0.0/24 is subnetted, 1 subnets
B       10.10.10.0 [20/100] via 172.16.0.1, 00:01:44
```

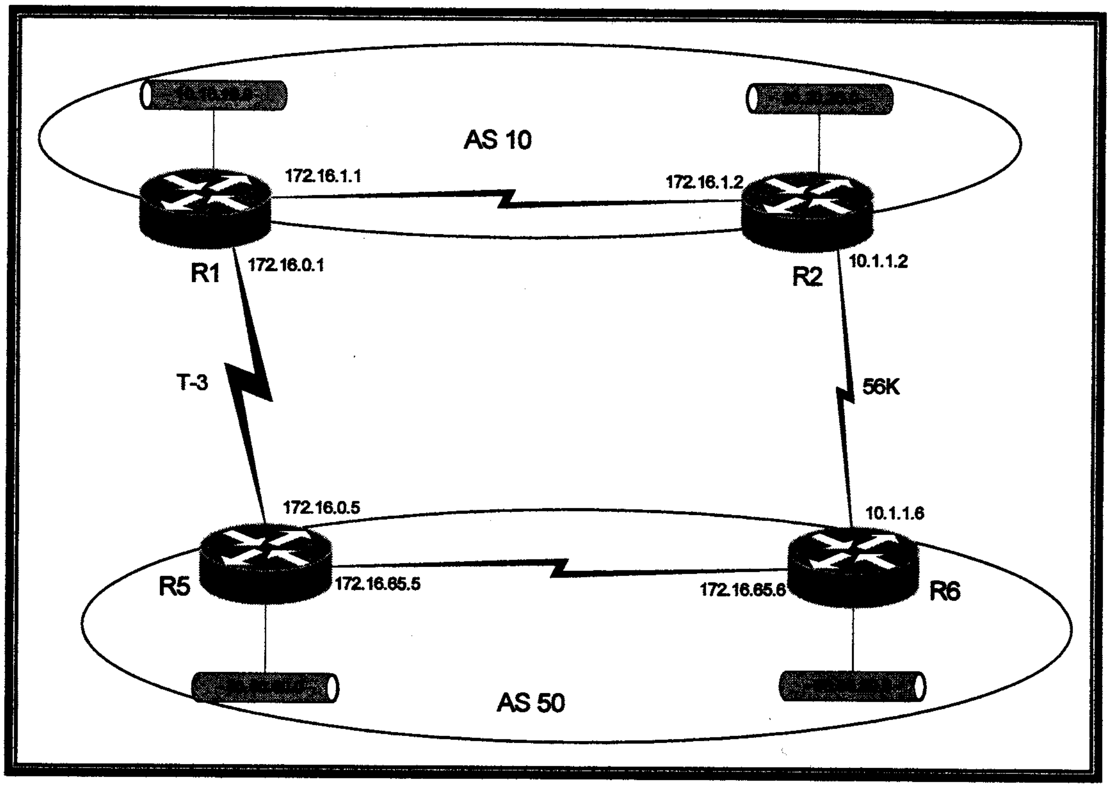
At this point, you should realize that configuring metrics really won't affect our paths. The reason is because AS path is a higher priority than metric and in our layout, the AS path is going to be the determining factor. However, if we had another network advertised to AS 50 that had the same path length via either AS 10 or AS 20 then the metric would be the deciding factor.

MED CONFIGURATION

The only difference between configuring a metric to be used inside an AS and a MED is the direction on the neighbor statement. Instead of applying the route-map to incoming updates, we apply it to outgoing updates. However, there is a lot more preparation and planning involved when you want to influence the routing of another AS. In a production environment, this is something you should do with caution.

To demonstrate an example of MED's, we are going to change our network layout. R1 and R2 are now both in AS 10 as shown in Figure 8.6. Also, notice we added another router (R6) and network (60.60.60.0) to AS 50. There is an IGP running so all routers know how to reach the serial links between the routers. Synchronization is also disabled.

Figure 8.6 *BGP and MED's*



The purpose of this example is to demonstrate how to influence the routing of another AS. We want traffic destined from AS 10 to 50.50.50.0 to use the T3 link. We want traffic destined from AS 10 to 60.60.60.0 to use the 56k link.

Prior to configuring the MED's, check your BGP table so you can see how the routing works before you begin. Notice that on R1, the best path to both 50.50.50.0 and 60.60.60.0 is via 172.16.0.5 (R5). This is because an external path is preferred over an internal path.

```
r1# show ip bgp
BGP table version is 11, local router ID is 1.1.1.1
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
*> 10.10.10.0/24    0.0.0.0                  0         32768 i
*>i20.20.20.0/24    172.16.1.2               0    100      0 i
* i50.50.50.0/24    10.1.1.6                      100      0 50 i
*>                  172.16.0.5               0             0 50 i
*> 60.60.60.0/24    172.16.0.5                             0 50 i
* i                 10.1.1.6                 0    100      0 50 i

r2# show ip bgp
BGP table version is 12, local router ID is 2.2.2.2
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
*>i10.10.10.0/24    172.16.1.1               0    100      0 i
```

```
*> 20.20.20.0/24    0.0.0.0                  0         32768 i
*> 50.50.50.0/24    10.1.1.6                               0 50 i
* i                 172.16.0.5               0    100      0 50 i
* i60.60.60.0/24    172.16.0.5                    100      0 50 i
*>                  10.1.1.6                 0             0 50 i
```

Now that we understand how our routing is currently configured, we can change the MED.

Step 1 Configure the access-list to have the metrics applied.

```
r5(config)# access-list 10 permit 50.50.50.0 0.0.0.255
r5(config)# access-list 20 permit 60.60.60.0 0.0.0.255

r6(config)# access-list 10 permit 50.50.50.0 0.0.0.255
r6(config)# access-list 20 permit 60.60.60.0 0.0.0.255
```

Step 2 Configure the route-maps on R5 to set the metrics for the routes. The route-map SETMED1 is used to apply metrics to outbound networks from R5 to R1. We want the network 50.50.50.0 to be preferred on the T-3 link so make sure it has a lower metric than the one advertised by R6. R5 sets the metric for 50.50.50.0 at 50 and R6 sets it at 75. The path through R5 should be preferred for this network

```
r5(config)# route-map SETMED1 permit 10
r5(config-route-map)# match ip add 10
r5(config-route-map)# set metric 50

r5(config)# route-map SETMED1 permit 20
r5(config-route-map)# match ip add 20
r5(config-route-map)# set metric 100
```

Step 3 Configure the route-maps on R6 to set the metrics for the routes. The route-map SETMED2 is used to apply metrics to outbound networks from R6 to R2. We want the network 60.60.60.0 to be preferred on the 56K link so make sure it has a lower metric than the one advertised by R5. R6 sets the metric for 60.60.60.0 at 25 and R5 sets it at 100. The path through R6 should be preferred for this network

```
r6(config)# route-map SETMED2 permit 10
r6(config-route-map)# match ip add 10
r6(config-route-map)# set metric 75

r6(config)# route-map SETMED2 permit 20
r6(config-route-map)# match ip add 20
r6(config-route-map)# set metric 25
```

Step 4 Apply the route-maps to the neighbors.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 route-map SETMED1 out

r6(config)# router bgp 50
r6(config-router)# neighbor 10.1.1.2 route-map SETMED2 out
```

MED Verification

To verify your routes, check the `show ip bgp` and `show ip route` command on both R1 and R2. You should notice that R2 sends traffic destined for 50.50.50.0 via R1. There are two different paths available for 60.60.60.0, but the one with the lower metric has the ">" signifying it is the best route.

```
r1# show ip bgp
BGP table version is 5, local router ID is 1.1.1.1
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
*> 10.10.10.0/24    0.0.0.0                  0         32768 i
*>i20.20.20.0/24    172.16.1.2               0    100      0 i
*> 50.50.50.0/24    172.16.0.5              50             0 50 i
*  60.60.60.0/24    172.16.0.5             100             0 50 i
*>i                 10.1.1.6                25    100      0 50 i

r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
     50.0.0.0/24 is subnetted, 1 subnets
B       50.50.50.0 [20/50] via 172.16.0.5, 00:15:08
     20.0.0.0/24 is subnetted, 1 subnets
B       20.20.20.0 [200/0] via 172.16.1.2, 00:14:37
     172.16.0.0/24 is subnetted, 3 subnets
C       172.16.0.0 is directly connected, Serial0
C       172.16.1.0 is directly connected, Serial1
O       172.16.65.0 [110/128] via 172.16.0.5, 00:16:10, Serial0
     10.0.0.0/24 is subnetted, 2 subnets
O       10.1.1.0 [110/128] via 172.16.1.2, 00:16:11, Serial1
C       10.10.10.0 is directly connected, Ethernet0
     60.0.0.0/24 is subnetted, 1 subnets
B       60.60.60.0 [200/25] via 10.1.1.6, 00:14:34
```

R1 sends traffic destined for 60.60.60.0 via R2. There are two different paths available for 50.50.50.0, but the one with the lower metric has the ">" signifying it is the best route.

```
r2# show ip bgp
BGP table version is 5, local router ID is 2.2.2.2
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
*>i10.10.10.0/24    172.16.1.1               0    100      0 i
*> 20.20.20.0/24    0.0.0.0                  0         32768 i
*  50.50.50.0/24    10.1.1.6                75             0 50 i
*>i                 172.16.0.5              50    100      0 50 i
*> 60.60.60.0/24    10.1.1.6                25             0 50 i

r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
B       50.50.50.0 [200/50] via 172.16.0.5, 00:17:18
     2.0.0.0/24 is subnetted, 1 subnets
C       2.2.2.0 is directly connected, Loopback0
     20.0.0.0/24 is subnetted, 1 subnets
```

```
C       20.20.20.0 is directly connected, Ethernet0
     172.16.0.0/24 is subnetted, 3 subnets
O       172.16.0.0 [110/128] via 172.16.1.1, 00:18:52, Serial1
C       172.16.1.0 is directly connected, Serial1
O       172.16.65.0 [110/192] via 172.16.1.1, 00:18:52, Serial1
     10.0.0.0/24 is subnetted, 2 subnets
B       10.10.10.0 [200/0] via 172.16.1.1, 00:17:20
C       10.1.1.0 is directly connected, Serial0
     60.0.0.0/24 is subnetted, 1 subnets
B       60.60.60.0 [20/25] via 10.1.1.6, 00:17:20
```

Route Aggregation and Auto Summary

By default, Cisco routers running BGP also auto summarize on the major classful boundary. Unless your AS is authority for that entire classful network, it is highly advised that you disable auto summary.

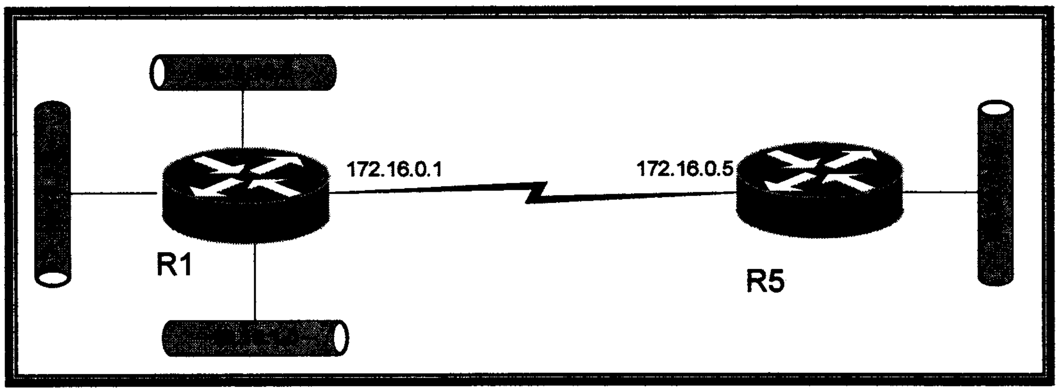
```
r5(config)# router bgp 50
r5(config-router)# no auto-summary
```

Aggregation should be used whenever possible. This helps keep routing tables small and helps to prevent flapping when individual networks go down. Flapping can also be minimized with route dampening as discussed later in this chapter. As with any summary, make sure that you only include networks that you have administrative control over. Do not summarize networks that you do not control. This is not an issue in a lab environment, but it can be devastating on the Internet.

Configuring Aggregate Routes and Specific Routes

Figure 8.7 illustrates a BGP router that has several networks that can be aggregated.

Figure 8.7 *BGP Aggregation*



Prior to configuring aggregate routes, check your routing table to make sure you have at least one more specific route that matches the aggregate. For example, if you want to aggregate the 10.10.0.0 \16 you need at least one specific network that begins with 10.10.x.x such as 10.10.1.0 or 10.10.2.0, etc.

```
r1(config)# router bgp 50
r1(config-router)# aggregate-address 10.10.0.0 255.255.0.0
```

The aggregate as well as the specific routes now show up on R5's routing table.

```
r5# show ip bgp
BGP table version is 12, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.0.0/16     172.16.0.1                             0 10 i
*> 10.10.1.0/24     172.16.0.1               0             0 10 i
*> 10.10.2.0/24     172.16.0.1               0             0 10 i
*> 10.10.10.0/24    172.16.0.1               0             0 10 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial1.1
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B       10.10.0.0/16 [20/0] via 172.16.0.1, 00:00:46
B       10.10.1.0/24 [20/0] via 172.16.0.1, 00:00:46
B       10.10.2.0/24 [20/0] via 172.16.0.1, 00:00:46
B       10.10.10.0/24 [20/0] via 172.16.0.1, 00:00:46
```

CONFIGURING AGGREGATE ROUTES WITHOUT MORE SPECIFIC ROUTES

When you configure an aggregate route, you can specify that specific routes should be suppressed. In other words, if we are aggregating the 10.10.0.0 \16 network then we do not want individual networks such as 10.10.1.0 and 10.10.2.0, etc. to be sent to our neighbor AS. This can defeat the purpose of having the aggregate in the first place.

```
r1(config-router)# aggregate-address 10.10.0.0 255.255.0.0 summary-only

r5# show ip bgp
BGP table version is 7, local router ID is 5.5.5.5
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.0.0/16     172.16.0.1                             0 10 i
*> 50.50.50.0/24    0.0.0.0                  0         32768 i
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
```

```
C       50.50.50.0 is directly connected, Ethernet0
     5.0.0.0/24 is subnetted, 1 subnets
C       5.5.5.0 is directly connected, Loopback0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial1.1
     10.0.0.0/16 is subnetted, 1 subnets
B       10.10.0.0 [20/0] via 172.16.0.1, 00:12:40
```

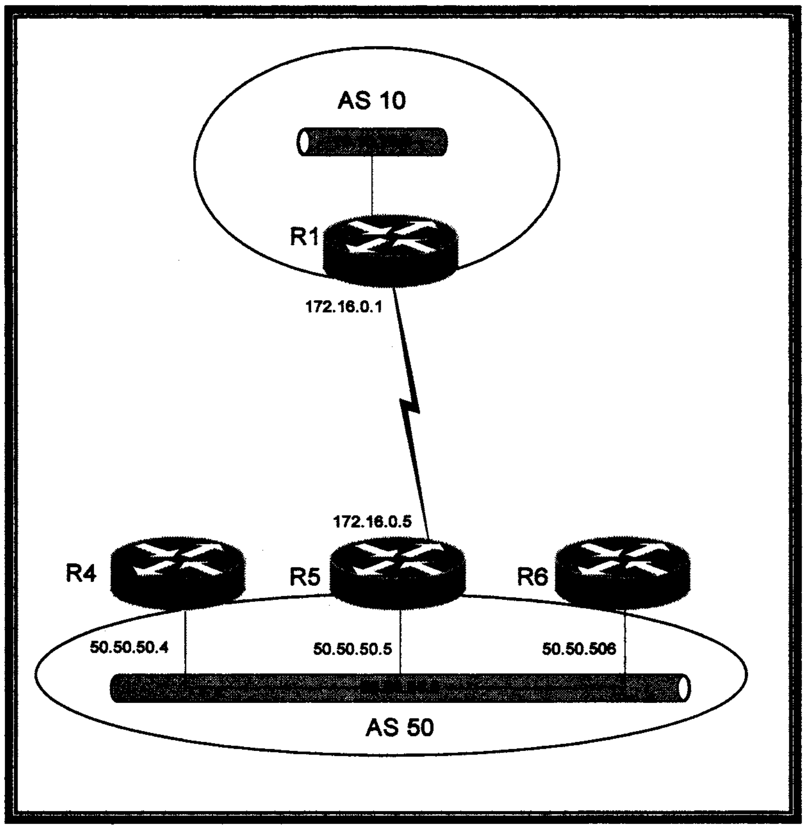
Route Reflectors

Autonomous systems consisting of hundreds of routing nodes can pose a serious routing management problem for network administrators. To make it easier to manage large numbers of BGP routers you can use Confederations or Route Reflectors.

Route reflectors reduce the number of IBGP peers by using a central router or "servers" that all the IBGP routers in the network communicate with. The clients then only need to maintain a connection to the server(s). This type of configuration can drastically reduce the number of BGP peers within your network. IBGP routers must be fully meshed so maintaining a large number of IBGP sessions may not be scalable.

In Figure 8.8., R5 will be the route reflector server. R4 and R6 will be route reflector clients. R4 and R6 do not need to have a neighbor relationship. Instead of having two neighbors, R4 and R6 now only have one, which results in half as many neighbor relationships to maintain. Imagine if there were dozens of routers on this same network. That's when route reflectors become very handy.

Figure 8.8 *BGP route reflectors*



Configuring route reflectors only requires a change on the server. The clients are not aware that they are not fully meshed. Any routes received from R4 will be "reflected" to R6 and vice versa. R4 and R6 do not have to be configured as peers using this setup.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 50.50.50.4 remote-as 50
r5(config-router)# neighbor 50.50.50.6 remote-as 50
r5(config-router)# neighbor 50.50.50.6 route-reflector-client
r5(config-router)# neighbor 50.50.50.4 route-reflector-client
```

ROUTE REFLECTOR VERIFICATION

To verify if your neighbors are configured properly simply enter the **show ip neighbor <address>** command.

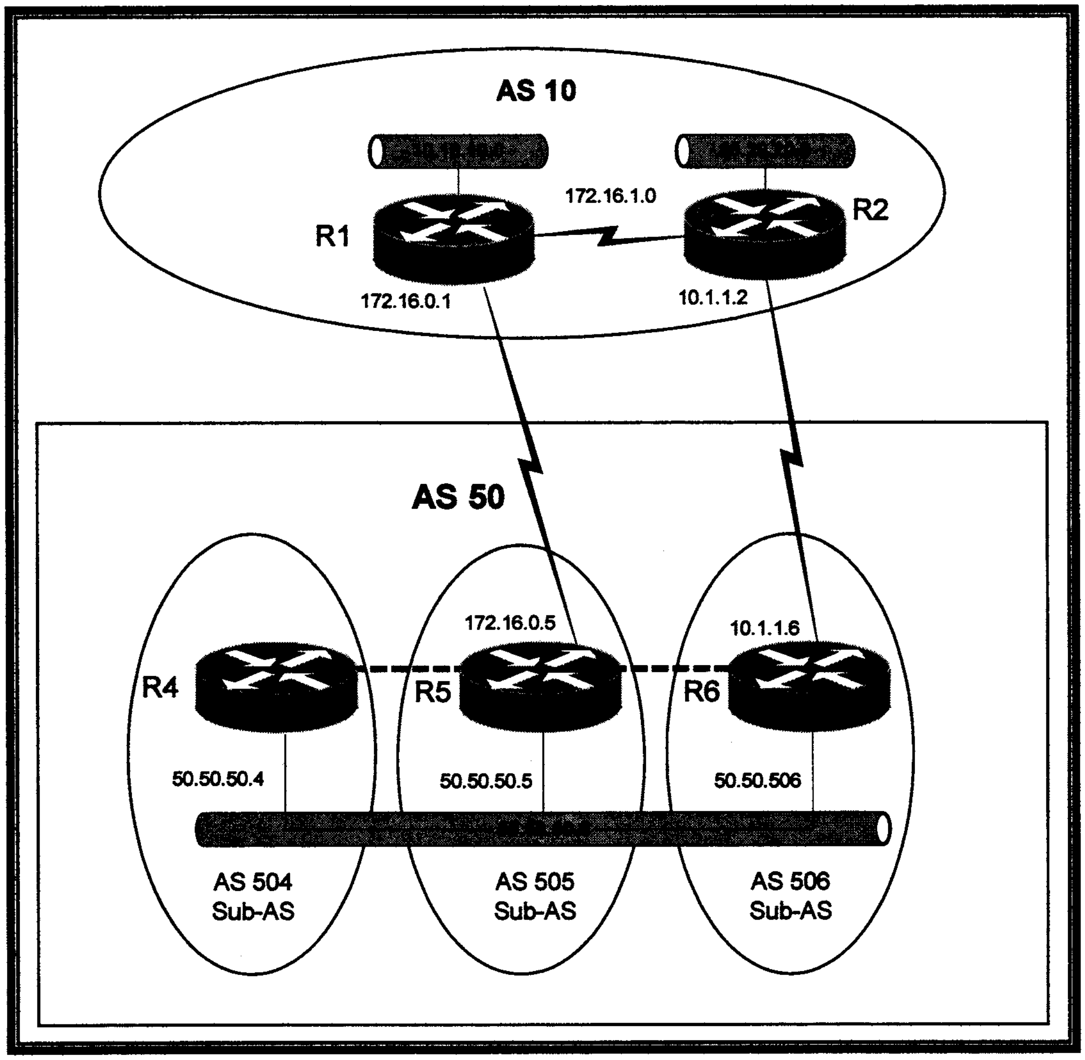
```
r5# show ip bgp neighbor 50.50.50.4
BGP neighbor is 50.50.50.4,  remote AS 50, internal link
 Index 3, Offset 0, Mask 0x8
  Route-Reflector Client
  BGP version 4, remote router ID 50.50.50.4
  BGP state = Established, table version = 8, up for 00:03:07
  Last read 00:00:07, hold time is 180, keepalive interval is 60 seconds
  Minimum time between advertisement runs is 5 seconds
  Received 8 messages, 0 notifications, 0 in queue
  Sent 9 messages, 0 notifications, 0 in queue
  Prefix advertised 5, suppressed 0, withdrawn 0
  Connections established 1; dropped 0
  Last reset 00:03:17, due to User reset
  1 accepted prefixes consume 32 bytes
  0 history paths consume 0 bytes
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 50.50.50.50, Local port: 11013
Foreign host: 50.50.50.4, Foreign port: 179

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0xB18818):
Timer          Starts    Wakeups            Next
Retrans            10          0             0x0
TimeWait            0          0             0x0
AckHold             7          5             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 4231181874  snduna: 4231182184  sndnxt: 4231182184     sndwnd:  16075
irs:  713299903  rcvnxt:  713300118  rcvwnd:      16170  delrcvwnd:    214

SRTT: 484 ms, RTTO: 3284 ms, RTV: 1158 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 300 ms, ACK hold: 200 ms
Flags: higher precedence, nagle

Datagrams (max data segment is 1460 bytes):
Rcvd: 12 (out of order: 0), with data: 7, total data bytes: 214
Sent: 16 (retransmit: 0), with data: 9, total data bytes: 309
```

The group of clients and the server is considered a "cluster." When there are multiple route reflector servers in a cluster, you must configure a common cluster ID on all routers using the **bgp cluster-id** command. This applies to servers and clients.

CONFEDERATIONS

Confederations reduce the number of IBGP peers by splitting a single AS into sub-AS's and using EBGP between them. Normally, private AS numbers are used for sub-AS's (AS numbers 64512 through 65536 are available for private use similar to the way RFC 1918 private IP addresses are available for internal use). In a large BGP environment, it is not scalable to have all BGP routers peered to each other. To external AS's, the entire confederation group looks like a single AS.

Figure 8.9 *BGP Confederations*



The routers in AS 10 will be configured normally. They should be completely unaware of the confederation. First, divide the main AS in to sub-AS's as shown in Figure 8.9. R4 and R6 are both neighbors to R5. This allows R4 and R6 to only have one neighbor configured thereby solving the same problem we had with route reflectors.

To configure confederations, you need to remove the current BGP configuration. Then start a new BGP configuration using the sub-AS. The confederation identifier is the main AS. Then configure the AS's that are peered. Remember that R5 is peered with both R4 and R6.

```
r5(config)# router bgp 505
r5(config-router)# bgp confederation identifier 50
r5(config-router)# bgp confederation peers 504 506

r4(config)# router bgp 504
r4(config-router)# bgp confederation identifier 50
r4(config-router)# bgp confederation peers 505
r6(config)# router bgp 506
r6(config-router)# bgp confederation identifier 50
r6(config-router)# bgp confederation peers 505
```

Configure the remaining BGP tasks as you normally would. But, remember to use the sub-AS numbers in your remote-as commands.

```
r5(config-router)# network 50.50.50.0 mask 255.255.255.0
r5(config-router)# neighbor 50.50.50.4 remote-as 504
r5(config-router)# neighbor 50.50.50.6 remote-as 506

r4(config-router)# neighbor 50.50.50.5 remote-as 505

r6(config-router)# neighbor 50.50.50.5 remote-as 505
```

Confederation Verification

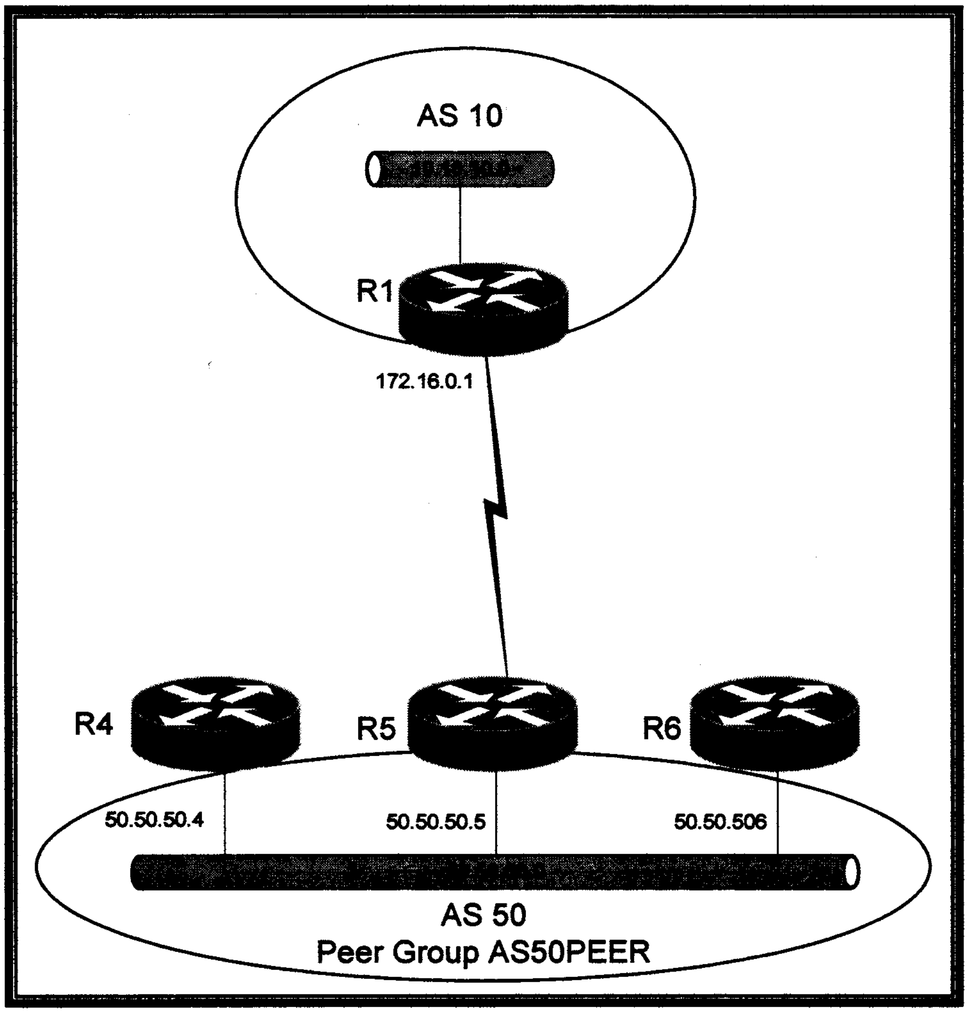
The **show ip bgp** command will show that the routes are learned via AS 505. On R6, it will also learn routes from its connection to AS 10.

```
r4# show ip bgp
BGP table version is 4, local router ID is 50.50.50.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.0.0/16     172.16.0.1                    100      0 (505) 10 i
*> 20.20.20.0/24    172.16.0.1                    100      0 (505) 10 i
*> 50.50.50.0/24    50.50.50.5               0    100      0 (505) i

r6# show ip bgp
BGP table version is 12, local router ID is 172.16.65.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.10.0.0/16     10.1.1.2                               0 10 i
*                   172.16.0.1                    100      0 (505) 10 i
*> 20.20.20.0/24    10.1.1.2                 0             0 10 i
*                   172.16.0.1                    100      0 (505) 10 i
*> 50.50.50.0/24    50.50.50.5               0    100      0 (505) I
```

BGP Peer Groups

A BGP peer group is a defined group of BGP neighbors that are configured to share the same update policies. Instead of defining the same policies for each individual neighbor, you can define a peer group name and assign policies to the peer group itself. Each member of the group will inherit these policies unless they are configured for a different policy. However, only inbound policies can be overridden. Outbound policies defined by the peer group cannot be changed. For example, a router can have a different set of filter-lists for inbound traffic. But, filter-lists for outbound traffic applied to the peer-group must be used by all routers.

Figure 8.10 *BGP Peer Groups*



Configuring peer groups has two required commands and several optional commands.

Step 1 Configure the peer-group and assign the group to an AS.

```
r5(config-router)# neighbor AS50PEER peer-group
r5(config-router)# neighbor AS50PEER remote-as 50
```

Step 2 Apply the neighbors to the peer-group created in the previous step.

```
r5(config-router)# neighbor 50.50.50.4 peer-group AS50PEER
r5(config-router)# neighbor 50.50.50.6 peer-group AS50PEER
```

Step 3 Configure the optional parameters for the peer-group. There are several optional commands including update-source, filter-lists, route-map, password, etc. See the command line help below as well as a two examples.

```
r5(config-router)# neighbor AS50PEER ?
  advertise-map             specify route-map for conditional advertisement
  advertisement-interval  Minimum interval between sending EBGP routing
updates
  default-originate         Originate default route to this neighbor
  description               Neighbor specific description
  distribute-list           Filter updates to/from this neighbor
  ebgp-multihop             Allow EBGP neighbors not on directly connected
                            networks
  filter-list               Establish BGP filters
  maximum-prefix            Maximum number of prefix accept from this peer
  next-hop-self             Disable the next hop calculation for this
neighbor
  password                  Set a password
  peer-group                Configure peer-group
  prefix-list               Filter updates to/from this neighbor
  remote-as                 Specify a BGP neighbor
  remove-private-AS         Remove private AS number from outbound updates
  route-map                 Apply route map to neighbor
  route-reflector-client    Configure a neighbor as Route Reflector client
  send-community            Send Community attribute to this neighbor
  shutdown                  Administratively shut down this neighbor
  soft-reconfiguration      Per neighbor soft reconfiguration
  timers                    BGP per neighbor timers
  unsuppress-map            Route-map to selectively unsuppress suppressed
routes
  update-source             Source of routing updates
  version                   Set the BGP version to match a neighbor
  weight                    Set default weight for routes from this neighbor

r5(config-router)# neighbor AS50PEER update-source loopback0
r5(config-router)# neighbor AS50PEER filter-list 1 out
```

Route Dampening

Route dampening is a BGP feature that attempts to minimize the propagation of flapping routes (up, down, up, down, etc.) across a network. Penalties are assigned for each flap and when the accumulated penalty reaches a specified limit, BGP suppresses further advertisement of that route, even if it is up. The accumulated penalty is reduced over time and eventually the route is trusted again. If your network is connected to many AS's with flapping routes, this can become a major burden on your BGP routers as they continuously update their routing tables.

A route flap occurs any time the path information changes for a route. We configured R6 to dampen the 20.20.20.0 network received from R2. Access-lists are used to identify the network that we want to dampen. However, access-lists are not required as we have several options to configure dampening. It can be configured globally for the BGP process, on routes received from a particular AS, or on specific routes regardless of which AS they are received.

Step 1 Configure your access-list to match the routes that you want to dampen.

```
r5(config)# access-list 20 permit 20.20.20.0 0.0.0.255
```

Step 2 Configure the route-map and set the dampening policy. Whenever the route flaps it is given a penalty. By default, the penalty for each flap is 1000. If that penalty reaches the suppress value it is then damped.

The first value in dampening is half-life. The half-life is what determines how fast the penalty will decay or decrease. The lower the half-life, the quicker the penalty will decay back to zero. The default is 15 minutes. We want our penalty to decay quicker so we chose 5 minutes.

The second value is reuse limit. Once a route has been dampened, it cannot be advertised again until the penalty falls below the reuse limit. The default reuse limit is 750.
The third value is the suppress value. Our suppress value is 1000 so when the route flaps once it will be dampened. This is probably not a good solution for a production network, but for our practice lab, it is easy to generate a few flaps in order to cause a dampening condition. The default suppress value is 2500.

The last value is the maximum suppress time. This is the maximum amount of time that a route can be suppressed. This value can range from 1 to 255 and defaults to four times the half-life time.

```
r5(config)# route-map DAMPEN_20_20_20_0 permit 10
r5(config-route-map)# match ip address 20
r5(config-route-map)# set dampening 5 100 1000 20
```

Step 3 Apply the route-map to the BGP process.

```
r5(config)# router bgp 50
r5(config-router)# bgp dampening route-map DAMPEN_20_20_20_0
```

DAMPENING VERIFICATION

To test our dampening, we caused a flap with the ethernet interface on R2. After two flaps, R6 put the route in to the dampening condition.

```
r6# show ip bgp 20.20.20.0
BGP routing table entry for 20.20.20.0/24, version 14
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.1.1.2
  10
    172.16.0.1 from 5.5.5.5 (5.5.5.5)
      Origin IGP, localpref 100, valid, internal, best
  10, (suppressed due to dampening)
    10.1.1.2 from 10.1.1.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, external
      Dampinfo: penalty 958, flapped 2 times in 00:07:42, reuse in 00:16:00

r6# show ip bgp dampened-paths
BGP table version is 14, local router ID is 6.6.6.6
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          From              Reuse    Path
*d 20.20.20.0/24    10.1.1.2          00:16:00 10 i
```

Once the penalty fell below 100, the route was advertised again.

```
r6# show ip bgp 20.20.20.0
BGP routing table entry for 20.20.20.0/24, version 15
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  5.5.5.5
  10
    172.16.0.1 from 5.5.5.5 (5.5.5.5)
      Origin IGP, localpref 100, valid, internal
  10
    10.1.1.2 from 10.1.1.2 (2.2.2.2)
      Origin IGP, metric 0, localpref 100, valid, external, best
```

Soft Reconfiguration

Soft reconfiguration allows BGP policies to be changed without resetting the TCP session between peers. By now, you should have noticed that every time you make a change to BGP, you have to reset the session and wait for the session to be reestablished. Although inconvenient, the load on the router and the impact to the network is minimal in a lab environment. On a production network, frequent policy changes and consequent resetting of TCP connections can have a dramatic effect on the performance of the network.

Outbound Soft Reconfiguration

Outbound soft reconfiguration does not require any special configuration. Instead of entering **`clear ip bgp *`** use the **`clear ip bgp * soft out`** command. Alternatively, you can specify the neighbor as shown in the example below.

```
r5# clear ip bgp 172.16.0.1 soft out
```

Inbound Soft Reconfiguration

Unlike outbound soft reconfiguration, inbound requires configuration. Inbound updates are stored in memory without modification. When a BGP session is cleared, the updates stored in memory are also cleared. The drawback to this is the additional memory overhead.

```
r5(config)# router bgp 50
r5(config-router)# neighbor 172.16.0.1 soft-reconfiguration inbound

r5# clear ip bgp 172.16.0.1 soft in
```

TYPICAL GOTCHAS!

- Entering a network statement that does not EXACTLY match with what is in the routing table.
- Missing update-source when using anything but the physical interface
- Forgetting to ad EBGP multihop if not using the directly connected physical interfaces
- Not turning Synchronization off, unless specifically told not to
- Not configuring 'next-hop-self' if a IBGP neighbor has no route to the current next hop
- Forgetting that a Prefix-list has an implicit deny at the end just like Access-lists
- Losing the community tag by not configuring 'send-community' on all neighbors
- Remembering that with WEIGHT and LOCAL-PREFERENCE higher wins, with MED lower wins
- Adding an unreachable, or make-believe as-path, while configuring AS-PATH prepend
- Adding unneeded configuration to route reflector clients – only the route reflector itself get any extra configuration
- Forgetting confederation identifier and peer statements
  Remembering that external AS's only see your main AS; not your configured confederations

CHAPTER 9

EIGRP

EIGRP is a Cisco proprietary protocol that combines the advantages of link state and distance vector routing protocols. Its characteristics classify it as a "hybrid" routing protocol. It supports automatic route summarization and VLSM addressing.

EIGRP was designed to overcome scaling limitations of IGRP. This was achieved by implementing the following.

- The Diffusing Update Algorithm (DUAL)
- Loop-free networks
- Incremental updates
- The holding of information about neighbors as opposed to the entire network

FEATURES OF EIGRP

- *Neighbor Discovery/Recovery* - Routers dynamically learn of other routers on their directly attached networks by sending a *Hello Packet.* As long as the neighbor receives these packets the router is assumed to be "alive."
- *Reliable Transport* - Ordered delivery of EIGRP packets to neighbors is guaranteed. For better efficiency, reliable transport is provided only when it is needed.
- *DUAL (Diffusing Update Algorithm)* - Tracks all the routes advertised by all neighbors. DUAL will use the metric to select an efficient path. It selects routes to be inserted into the routing table based on feasible successors.
- *Protocol Dependent Modules* - These are responsible for the network layer. The IPX EIGRP module is responsible for sending and receiving EIGRP packets that are encapsulated in IPX. The Apple EIGRP module is responsible for AppleTalk packets.

TYPES OF SUCCESSORS

- *Successor* - A route selected as the primary route to use to reach a destination. Successors are the entries kept in the routing table.
- *Feasible Successor* - A backup route. Multiple feasible successors for a destination can be retained, kept in topology table. This way, when a route goes down, it does not have to recompute the entire routing table to enter the route into it's table.

Tables

- *Neighbor table* - The current state of all the router's immediately adjacent neighbors.
- *Topology table* - This table is maintained by the protocol dependent modules and is used by DUAL. It has all the destinations advertised by the neighbor routers.
- *Routing table* - EIGRP chooses the best routes to a destination from the topology table and places these routes in the routing table. The routing table contains:
  - How the route was discovered
  - Destination network address and the subnet mask
  - Administrative Distance which is the metric or cost from the neighbor advertising that particular route
  - Metric Distance which is the cost or the metric from the router
  - Next hop address
  - Route age
  - Outbound interface

Choosing routes

DUAL selects primary and backup routes based on the composite metric and ensures that the selected routes are loop free. The primary routes are then moved to a routing table. The rest (up to 6) are stored in the topology table.

EIGRP uses the same composite metric as IGRP to determine the best path. The default criteria used are listed below.

- *Bandwidth* - the smallest bandwidth cost between source and destination
- *Delay* - cumulative interface delay along the path
- *Reliability* - worst reliability between source and destination based on keepalives
- *Load* - utilization on a link between source and destination based on bits per second on its worst link
- *MTU* - the smallest Maximum Transmission Unit

Basic EIGRP configuration

Step 1 Configure EIGRP for IP. Enable EIGRP and define the autonomous system Remember that BGP, EIGRP and IGRP have AS numbers (OSPF has a process ID, RIP requires neither). In order for two EIGRP routers to exchange routes they must be in the same AS.

```
r4(config)# router eigrp 1
```

Step 2 Indicate which networks are part of the EIGRP autonomous system. If you do not specify a wildcard mask the router will automatically switch to the classful boundary. Then verify the router is routing for that network.

```
r4(config-router)# network 50.50.50.0
r4# sh ip protocol
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Routing for Networks:
    50.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

-OR-

```
r4(config-router)# network 50.50.50.0 0.0.0.255

r4# show ip protocol
01:06:06: %SYS-5-CONFIG_I: Configured from console by consoleo
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Routing for Networks:
    50.50.50.0/24
  Routing Information Sources:
    Gateway         Distance      Last Update
  Distance: internal 90 external 170
```

MANIPULATING ROUTES

EIGRP has several options to manipulate routing. You can change the default metrics, weights for each metric, summarize routes, load balance traffic, and create stub networks. Each of these options is further explained in the remainder of this chapter.

ADJUSTING EIGRP METRICS

EIGRP metrics consist of bandwidth, delay, load, and reliability. It is important to understand what each factor means when computing a metric.

If k5 equals 0, the EIGRP metric is computed according to the following formula:

metric = [(k1 * bandwidth) + (k2 * bandwidth)/(256 - load) + (k3 * delay) * (256)]

If k5 does not equal zero, an additional operation is performed:

metric = metric * [k5/(reliability + k4)]

Table 9.1 *EIGRP metric weights*

Constant	Default value	Relationship to EIGRP metric
K1	1	Bandwidth calculation
K2	0	Load calculation (if the value is 0, load has no effect on the metric)
K3	1	Delay calculation
K4	0	Reliability calculation
K5	0	Reliability calculation (if the value is 0, this metric is not included at all)

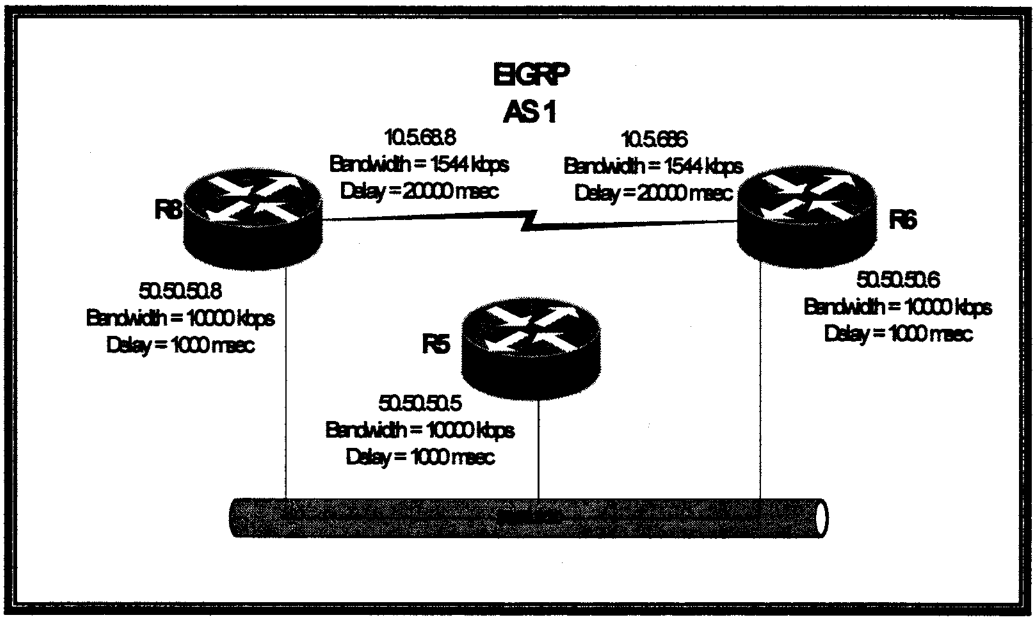
By default, load and reliability have no effect on the EIGRP metrics!

The current EIGRP metric weights are shown by entering the **show ip protocol** command. This will display the default metric weights unless they have been changed by the administrator.

```
r6# show ip protocol
Routing Protocol is "eigrp 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 1
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    50.0.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    50.50.50.4            90      00:40:33
    50.50.50.5            90      00:10:16
    50.50.50.8            90      00:15:02
  Distance: internal 90 external 170
```

Let's configure a small network as shown in Figure 9.2. Calculate the metric for the serial connection between R6 and R8 to be sure you understand exactly how the metric is computed.

Figure 9.1 *EIGRP weight example*



- Delay is in units of 10 microseconds and the calculation is <cumulative delay> / 10
- Bandwidth is N Kbps and the calculation is 10000000 / N
- Reliability is given as a fraction of 255 with 255 being 100% reliable
- Load is given as a fraction of 255 with 255 being fully loaded

Knowing the constant values is only half the actual metric. Now we need to plug in our values for bandwidth and delay. Remember that unless you change the constants, load and reliability are not factored in to the metric. The first task is to identify the total delay and minimum bandwidth. The **show ip route <network>** command will provide this information as shown below.

```
r5# show ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "eigrp 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 50.50.50.6 on Ethernet0, 00:00:04 ago
  Routing Descriptor Blocks:
  * 50.50.50.8, from 50.50.50.8, 00:00:04 ago, via Ethernet0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    50.50.50.6, from 50.50.50.6, 00:02:26 ago, via Ethernet0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

Bandwidth is not cumulative. Its value is simply the smallest link along the path from the router to the destination. If there was a low speed, 56K serial line between R5 and R6, the minimum bandwidth for the route via R6, would only be 56Kbit instead of 1544Kbit.

Delay is the total value for each segment in the path. R5 only traverses two links to reach the 10.0.0.0 network regardless of which path it takes. R5's Ethernet 0 has a delay of 1000 msec and R6's Serial 0 has a delay of 20000 msec.

Load is how much of that bandwidth is currently in use. This value is indicated as a fraction. If there is no load, the value will be 1/255.

Reliability indicates the current error rate. This value is indicated as a fraction. If the link is clean, the value will be 255/255.

To check the bandwidth, delay, reliability, and load of an interface simply enter the **show interface** command.

```
r5# show interface ethernet 0
Ethernet0 is up, line protocol is up
  Hardware is Lance, address is 0000.0c31.9fcf (bia 0000.0c31.9fcf)
  Internet address is 50.50.50.5/24
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, loopback not set, keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:01, output 00:00:01, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     1147 packets input, 92716 bytes, 0 no buffer
     Received 861 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 input packets with dribble condition detected
     1052 packets output, 78555 bytes, 0 underruns
     0 output errors, 0 collisions, 16 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

We still need to compute the actual values for bandwidth and delay. Then we can plug all of our values in to the metric. Note that the bandwidth value is rounded down and not up. Also, remember that since K5 equals 0, reliability is not factored in to the metric. The next three sections break down how we compute the metric based on the values and the equation.

VALUES

K1 = 1
K2 = 0
K3 = 1
K4 = 0
K5 = 0
Bandwidth = 6476.6839378 (1000000 / 1544)
Delay = 2100 (21000 / 10)

EQUATION

[(k1 * bandwidth) + (k2 * bandwidth)/(256 - load) + (k3 * delay) * (256)]

COMPUTATION

Enter all the values.

[(1 * 6476) + (0*6476)/(256-1) + (1 * 2100) * (256)]

Compute the equation.

[(6476) + (0/255) + (2100) * 256] = 2195456

```
r5# sh ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "eigrp 1", distance 90, metric 2195456, type internal
  Redistributing via eigrp 1
  Last update from 50.50.50.6 on Ethernet0, 02:56:36 ago
  Routing Descriptor Blocks:
  * 50.50.50.8, from 50.50.50.8, 02:56:36 ago, via Ethernet0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    50.50.50.6, from 50.50.50.6, 02:56:36 ago, via Ethernet0
      Route metric is 2195456, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

Now that we understand how EIGRP computes its metrics, we can change a weight value to put more emphasis on one of the metric weights. In our example, we are giving bandwidth more emphasis when computing a metric by doubling its weight to 2. The first "0" is type of service and 0 is the only supported value. The remaining values are K1 through K5 in order.

```
r5(config)# router eigrp 1
r5(config-router)# metric weights 0 2 0 1 0 0
```

When we show our route, we see that the metric has in fact doubled as we expected.

```
r5# show ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "igrp 1", distance 100, metric 15052
  Redistributing via igrp 1
  Last update from 50.50.50.6 on Ethernet0, 00:00:02 ago
  Routing Descriptor Blocks:
  * 50.50.50.8, from 50.50.50.8, 00:00:03 ago, via Ethernet0
      Route metric is 15052, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 0
    50.50.50.6, from 50.50.50.6, 00:00:02 ago, via Ethernet0
      Route metric is 15052, traffic share count is 1
      Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 0
```

A reasonable question on the exam may be to add reliability and load values to the metric. Notice the new metric.

```
r5(config)# router eigrp 1
r5(config-router)# metric weights 0 1 1 1 1 1

r5# sh ip route 10.0.0.0
Routing entry for 10.0.0.0/8
  Known via "eigrp 1", distance 90, metric 8601, type internal
  Redistributing via eigrp 1
  Last update from 50.50.50.8 on Ethernet0, 00:00:07 ago
  Routing Descriptor Blocks:
```

```
* 50.50.50.6, from 50.50.50.6, 00:00:07 ago, via Ethernet0
    Route metric is 8601, traffic share count is 1
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
  50.50.50.8, from 50.50.50.8, 00:00:07 ago, via Ethernet0
    Route metric is 8601, traffic share count is 1
    Total delay is 21000 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
```

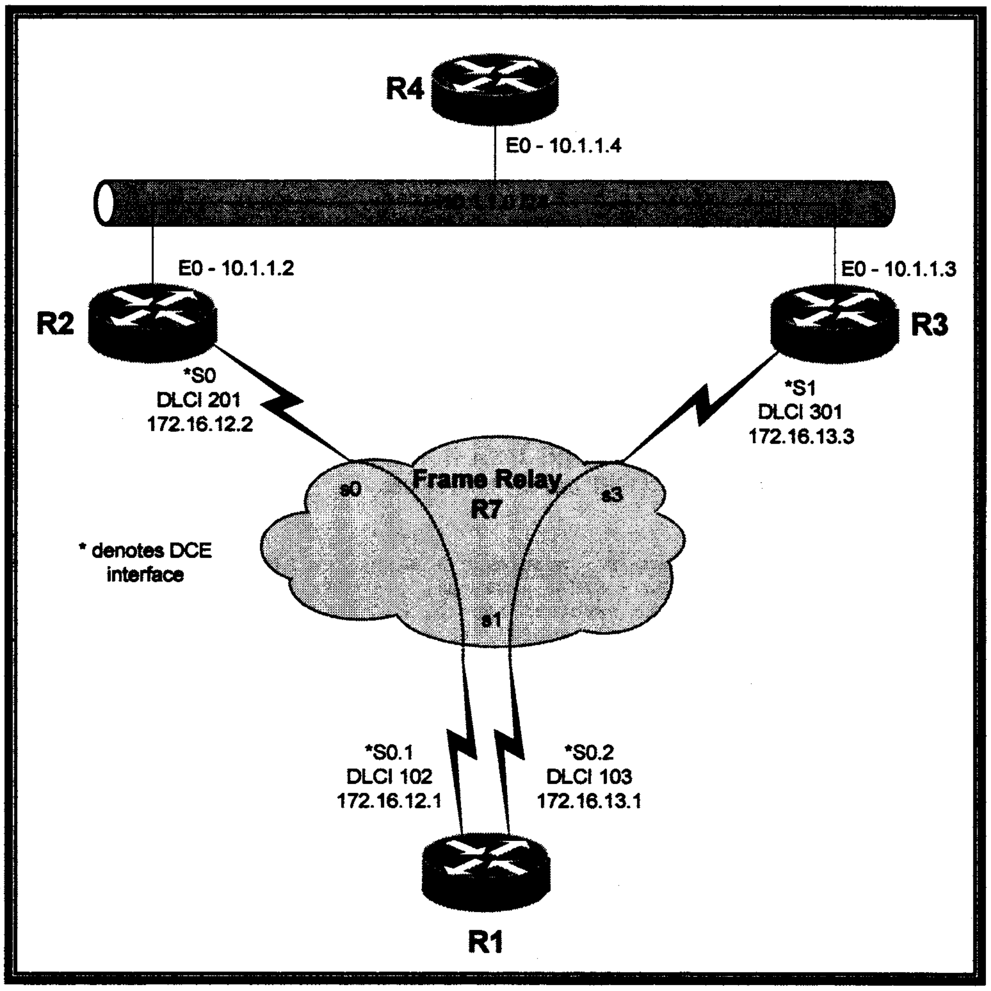
Manipulating Default Metrics

By default, EIGRP only uses bandwidth and delay as its metrics. Both metrics are easily changed. However, if you have OSPF or another routing protocol that uses bandwidth, you may inadvertently change the metric for that routing protocol as well. For that reason you want to be sure you know how to change either metric.

In figure 9.1, we configured a basic EIGRP network. R1 will reach the 10.1.1.0 network using both R2 and R3 since the current bandwidth is the same through either interface.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
D    10.0.0.0/8 [90/40537600] via 172.16.13.3, 00:20:09, Serial0.2
                [90/40537600] via 172.16.12.2, 00:20:09, Serial0.1
```

Figure 9.2. *EIGRP Metric Manipulation*



We set both interfaces to 64 kbps. Once we changed the bandwidth on s0.1 to 128 kbps, the route through R3 drops out of the routing table. It is still in the topology table, however it is not a successor.

```
r1(config)# interface serial 0.1
r1(config-subif)# bandwidth 128

r1# clear ip route *
r1# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
```

```
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
D    10.0.0.0/8 [90/20537600] via 172.16.12.2, 00:00:02, Serial0.1

r1# show ip eigrp topology
IP-EIGRP Topology Table for AS(1)/ID(172.16.13.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status

P 10.0.0.0/8, 1 successors, FD is 2195456
         via 172.16.12.2 (20537600/281600), Serial0.1
         via 172.16.13.3 (40537600/281600), Serial0.2
P 172.16.12.0/24, 1 successors, FD is 2169856
         via Connected, Serial0.1
P 172.16.13.0/24, 1 successors, FD is 2169856
         via Connected, Serial0.2
r1#
```

If we were also running OSPF on R1, R2, and R3 we would have just changed the routing for those networks as well. To illustrate, we configured OSPF on R1, R2, R3, and R4. We added one loopback on R4 using the network 192.168.4.4. When the bandwidth statements are both 64 kbps on the subinterfaces on R1, OSPF can use both routes.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
     192.168.4.0/32 is subnetted, 1 subnets
O IA    192.168.4.4 [110/1573] via 172.16.12.2, 00:00:02, Serial0.1
                    [110/1573] via 172.16.13.3, 00:00:02, Serial0.2
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.0.0.0/8 [90/40537600] via 172.16.13.3, 00:00:02, Serial0.2
                   [90/40537600] via 172.16.12.2, 00:00:02, Serial0.1
O IA    10.1.1.0/24 [110/1572] via 172.16.12.2, 00:00:03, Serial0.1
                    [110/1572] via 172.16.13.3, 00:00:03, Serial0.2
```

Assume that you are asked to prefer the route to 10.1.1.0 without changing OSPF's route to 192.168.4.4. After we change the serial 0.1 bandwidth to 128, we lose our other route to 192.168.4.4.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
```

```
     192.168.4.0/32 is subnetted, 1 subnets
O IA    192.168.4.4 [110/792] via 172.16.12.2, 00:00:02, Serial0.1
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.0.0.0/8 [90/40537600] via 172.16.12.2, 00:00:02, Serial0.1
                   [90/40537600] via 172.16.13.3, 00:00:02, Serial0.2
O IA    10.1.1.0/24 [110/791] via 172.16.12.2, 00:00:03, Serial0.1
```

Changing the delay is the only way to manipulate the default EIGRP metrics without changing the bandwidth. To check the delay for an interface, use the show interface command. The example below shows that OSPF maintains its equal cost routing and EIGRP prefers the route through R2.

```
r1(config)# interface serial 0.1
r1(config-subif)# delay 1500

r1# show interface serial 0.1
Serial0.1 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.16.12.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 15000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY

r1# show interface serial 0.2
Serial0.2 is up, line protocol is up
  Hardware is HD64570
  Internet address is 172.16.13.1/24
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY

r1# clear ip route *
r1# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
     192.168.4.0/32 is subnetted, 1 subnets
O IA    192.168.4.4 [110/1573] via 172.16.12.2, 00:00:01, Serial0.1
                    [110/1573] via 172.16.13.3, 00:00:01, Serial0.2
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    10.1.1.0/24 [110/1572] via 172.16.12.2, 00:00:01, Serial0.1
                    [110/1572] via 172.16.13.3, 00:00:01, Serial0.2
D       10.0.0.0/8 [90/40409600] via 172.16.12.2, 00:04:35, Serial0.1
```

Whenever you change the metrics, always make sure you change it on both sides of a link or you will have asymmetrical routing.

AUTO SUMMARIZATION

EIGRP automatically summarizes at the classful boundary. This may or may not be desirable. If you have discontiguous subnets you must disable it or your routing will not function properly. Automatic summarization can be disabled on the entire EIGRP process. Using the topology in figure 9.1, we disable auto-summary on all four routers. Prior to doing that, you should notice the 10.0.0.0 summary route on r1.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
D    10.0.0.0/8 [90/40537600] via 172.16.13.3, 00:00:01, Serial0.2
                [90/40537600] via 172.16.12.2, 00:00:01, Serial0.1

r1(config)# router eigrp 1
r1(config-router)# no auto-summary

r2(config)# router eigrp 1
r2(config-router)# no auto-summary

r3(config)# router eigrp 1
r3(config-router)# no auto-summary

r4(config)# router eigrp 1
r4(config-router)# no auto-summary

r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 2 subnets
C       172.16.12.0 is directly connected, Serial0.1
C       172.16.13.0 is directly connected, Serial0.2
     10.0.0.0/24 is subnetted, 1 subnets
D       10.1.1.0 [90/40537600] via 172.16.13.3, 00:00:06, Serial0.2
                 [90/40537600] via 172.16.12.2, 00:00:06, Serial0.1
```

MANUAL SUMMARIZATION

You can also configure specific summaries on a per-interface basis. There must be at least one specific route in order for EIGRP to advertise the summary. Refer back to figure 9.1 for the topology in this example. All the routers have auto-summary disabled. Now we configure R2 to send a summary on the frame-relay network. Now R1 has both the summary route and the more specific route from R3.

```
r2(config-if)# int serial 0
r2(config-if)# ip summary-address eigrp 1 10.0.0.0 255.0.0.0 5

r1# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.12.0/24 is directly connected, Serial0.1
C       172.16.13.0/24 is directly connected, Serial0.2
D       172.16.0.0/16 [90/41049600] via 172.16.13.3, 00:00:01, Serial0.2
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       10.0.0.0/8 [90/40537600] via 172.16.12.2, 00:00:00, Serial0.1
D       10.1.1.0/24 [90/40537600] via 172.16.13.3, 00:00:01, Serial0.2
```

Now that we understand how the summary works, let's throw in a CCIE type question. Configure your network so that R1 prefers the route through R3 without making any changes to any EIGRP metrics.

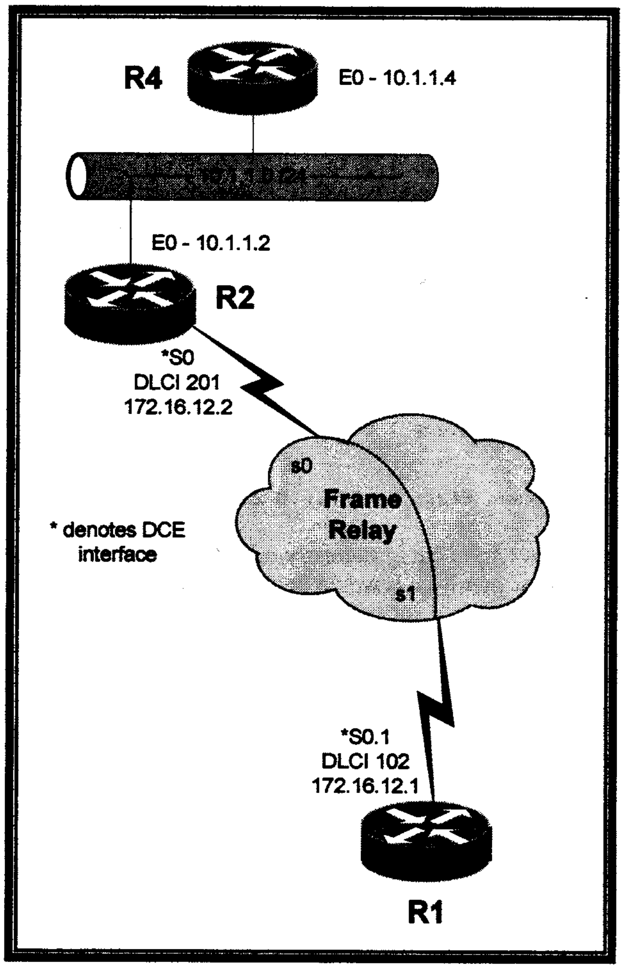
```
r3(config-if)# int serial 0
r3(config-if)# ip summary-address eigrp 1 10.0.0.0 255.0.0.0 5

r2(config-if)# int serial 0
r2(config-if)# ip summary-address eigrp 1 10.0.0.0 255.0.0.0 10
```

Using the administrative distance option within the ip summary-address command, we can manipulate the routing. Distance is not a metric so it would be a valid solution to the question.

Default Routing

EIGRP has default routing capability using the same ip summary command just discussed. As shown in figure 9.3, the only path to R4 is through R2.

Figure 9.3. *EIGRP Default Routing*



Another CCIE type question would be to have a default route appear on R4, but do not make any changes on R4 itself. You may not use any access-lists or filters on any router to accomplish this task. With OSPF and BGP this is easy because we have the default-information originate command. EIGRP can only originate a default route by using an all 0's summary.

```
r1(config-router)# int serial 0
r1(config-if)# ip summary-address eigrp 1 0.0.0.0 0.0.0.0

r4# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.1.1.2 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 1 subnets
```

```
D       172.16.12.0 [90/2195456] via 10.1.1.2, 00:17:48, Ethernet0
C    192.168.4.0/24 is directly connected, Loopback0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0
D*   0.0.0.0/0 [90/41049600] via 10.1.1.2, 00:00:22, Ethernet0
```

Default routing can be controlled with the default-information command under the EIGRP process. The different options are self-explanatory.

```
r4(config-router)# default-information ?
allowed  Allow default information
in       Accept default routing information
out      Output default routing information
```

Stub Routing

An EIGRP stub router can control which type of networks it will advertise. The important thing to remember about EIGRP stubs is that it only affects what the router sends, not what it will accept. Unlike OSPF, it does not filter any incoming routes nor does it create a default route.

Step 1 Identify the router(s) that will be configured as a stub. Only the remote router must be configured.

Step 2 Configure the stub router as a stub. There are four options: receive-only, connected, static, and summary.

```
r4(config)# router eigrp 1
r4(config-router)# eigrp stub summary
```

Can you think of the types of questions that they can ask you that would lead to these solutions? You should be able to. Here are a few examples.

Question 1: Configure R4 to not send any of its routes to its neighbors. You may not use any access-lists or change any administrative distances.

Answer 1: eigrp stub receive-only

Question 2: Advertise connected and static routes from R4, but do not use the redistribute command.

Answer 2: eigrp stub connected static

Question 3: Configure R4 to advertise a summary route. You must disable auto-summary and you cannot use the ip summary-address command on any interface.

Answer 3: eigrp stub summary

You can also check if your neighbor is configured as a stub. This may be helpful when connecting to other EIGRP routers that you do not control.

```
r3# show ip eigrp neighbor detail
IP-EIGRP neighbors for process 1
H   Address                 Interface   Hold Uptime   SRTT   RTO  Q  Seq Type
                                        (sec)         (ms)       Cnt Num
0   10.1.1.4                Et0           11 00:05:21  791  4746  0  77
   Version 12.1/1.2, Retrans: 0, Retries: 0
   Stub Peer Advertising ( SUMMARY ) Routes
2   172.16.13.1             Se1           13 02:36:11   43   258  0  287
   Version 12.1/1.2, Retrans: 15, Retries: 0
```

```
1   10.1.1.2              Et0           11 02:37:51   11   200  0  120
   Version 12.1/1.2, Retrans: 1, Retries: 0
```

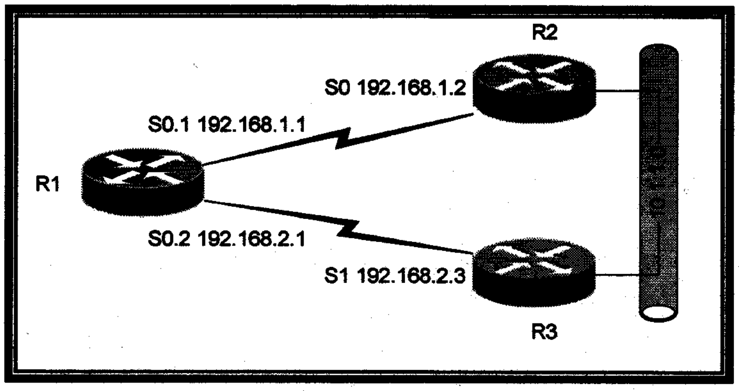
Unequal Cost Load Balancing

By default, EIGRP will load balance across four equal cost paths.

Offset-Lists

An offset-list can be used to manipulate the metrics of specific routes without adjusting the default EIGRP K value metrics. In Figure 9.4, R1 will use both R2 and R3 equally to reach the 10.2.2.0 network. Let's assume that R3 was a very busy router and we had an immediate need to bypass R3 and only use R2 to reach that network.

Figure 9.4 *EIGRP weight example*



```
r1# sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.1.1.0 [90/2195456] via 192.168.1.2, 00:02:07, Serial0.1
                 [90/2195456] via 192.168.2.3, 00:02:07, Serial0.2
C    192.168.1.0/24 is directly connected, Serial0.1
C    192.168.2.0/24 is directly connected, Serial0.2
```

We can use offset-lists to instruct R1 to prefer the route to 10.1.1.0 to use R2.

```
r1(config)# access-list 1 permit 10.1.1.0 0.0.0.255
r1(config)# router eigrp 1
r1(config-router)# offset-list 1 in 100000 serial0.1

r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
D       10.1.1.0 [90/2195456] via 192.168.2.3, 00:01:31, Serial0.2
C    192.168.1.0/24 is directly connected, Serial0.1
C    192.168.2.0/24 is directly connected, Serial0.2
```

Limiting EIGRP Bandwidth

The bandwidth used by EIGRP is limited to 50 percent by default. If the EIGRP updates are large this may cause problems on low spend links, on busy routers, or disparate link sizes typically found in hub-and-spoke frame-relay networks. The **ip bandwidth-percent eigrp** command is configured at the interface level.

```
r1(config)# interface s0
r1(config-if)# ip bandwidth-percent eigrp 25
```

Static Neighbors

Configuring EIGRP neighbors allows the router to send its routing updates via a unicast packet in addition to its regular multicast packets. EIGRP by default will send its updates to 224.0.0.10. Unlike other routing protocols, if you configure neighbor, the statements must be placed on all routers as shown below.

```
r4(config)# router eigrp 1
r4(config-router)# neighbor 50.50.50.5 Ethernet0
r4(config-router)# neighbor 50.50.50.6 Ethernet0

r5(config)# router eigrp 1
r5(config-router)# neighbor 50.50.50.4 Ethernet0
r5(config-router)# neighbor 50.50.50.6 Ethernet0

r6(config)# router eigrp 1
r6(config-router)# neighbor 50.50.50.4 Ethernet0
r6(config-router)# neighbor 50.50.50.5 Ethernet0
```

With some routing protocols, such as RIP, there is a broadcast issue with Frame-relay. EIGRP, however, can send and receive routes on a Frame-Relay network without configuring static neighbors. If you configure static neighbors, you must have a network statement configured for the network that the neighbor is found.

EIGRP Timers

Hello packets are transmitted every 5 seconds on links faster than T-1. NBMA links such as Frame-Relay and links slower than T-1 will transmit hello packets every 60 seconds. If an interface is configured to use physical mutlticasting it is not considered NBMA by EIGRP. The hello timers can be adjusted using the **ip hello-interval eigrp <as number> <seconds>** under the EIGRP process.

If a router does not converge quickly enough because links are too slow, the router is too busy, etc. there may be a "Stuck In Active" state which means the router believes that it is not completing the routing update processing in a timely manner. By default, the router will wait 3 minutes before generating this message. This timer can be configured your network frequently takes longer than 3 minutes. Changing these timers (or disabling it completely) would need to be done throughout your network.

```
r5(config-router)# timers active-time ?
  <1-4294967295>  EIGRP active-state time limit in minutes
  disabled        disable EIGRP time limit for active state

r5(config-router)# timers active-time 5
```

TYPICAL GOTCHAS!

- Remembering that the 1st digit in the 'metric weights' command is always 0 – TOS was never implemented
- Not matching the metric weights in all routers in the AS
- Forgetting to turn of auto-summary
- Remembering that a summary is done on the interface for EIGRP

CHAPTER 10

RIP

RIP is a classful protocol which means it sends all of its routes on major mask boundaries such as /8, /16, and /24. It does not send any subnet mask information with its routes. The metric used by RIP is hop count. It has timers similar to EIGRP and has a very slow convergence time.

BASIC RIP CONFIGURATION

First, configure the RIP routing process. Then assign a network or networks to the process.

```
r5(config)# router rip
r5(config-router)# network 172.16.0.0
r5(config-router)# network 50.0.0.0
```

ADJUSTING RIP TIMERS

RIP utilizes several timers in order to make routing table changes. The list below illustrates the configurable timers. These timers perform the same function as IGRP, but their default values are different.

- Update interval – how frequently the routing table updates are sent
- Invalid interval – route is unusable, but still in the routing table
- Hold down – route is still unusable and the route will not be accepted from any other routers
- Route removal interval - route is completely removed from the routing table

To check your current timers (default values if they haven't been adjusted) enter the **show ip protocol** command.

```
r5# show ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 15 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface           Send  Recv   Key-chain
    Ethernet0           1     1 2
    Serial1.1           1     1 2
  Routing for Networks:
    50.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.0.1           120      00:05:34
    172.16.0.2           120      00:06:27
  Distance: (default is 120)
```

RIP converges much more quickly than IGRP, but it still may not be fast enough for some environments. To speed up convergence, you can change the update interval to a smaller value such as 15 seconds as well as the other timer values. You should notice that the invalid timer is exactly 6 times the update interval. The hold down interval is the same as the invalid timer. The flush interval is exactly 8 times the update interval. It is probably easiest to continue to use these factors when adjusting the timers. The example below illustrates an RIP network that will converge twice as fast compared to using the default timers. Do not forget to use the same timers on all RIP routers in your network!

```
r5(config)# router rip
r5(config-router)# timers basic 15 90 90 120
```

There are several drawbacks to this approach. First, you need to make sure that all the timers match throughout your RIP network. In a large RIP network, this may be very time consuming and result in parts of your network being down until all routers are configured for the same timers. Second, this will cause a lot of RIP traffic since entire routing tables are sent every 15 seconds instead of 30 seconds. More bandwidth and router resources will be spent on RIP updates.

RIP Timers Verification

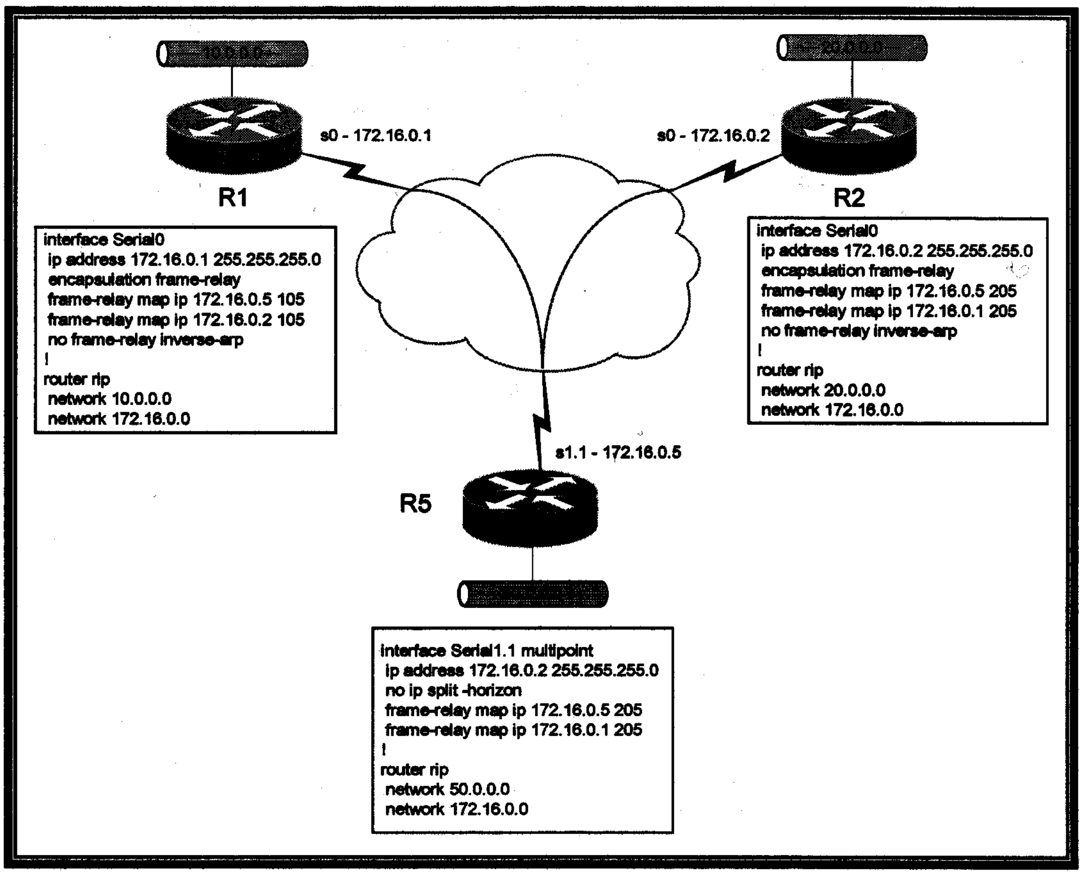
Use the `show ip protocol` command to check your current timers.

```
r5# show ip protocol
Routing Protocol is "rip"
  Sending updates every 15 seconds, next due in 5 seconds
  Invalid after 90 seconds, hold down 90, flushed after 120
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface             Send  Recv   Key-chain
    Ethernet0             1     1 2
    Serial1.1             1     1 2
  Routing for Networks:
    50.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway          Distance      Last Update
    172.16.0.1            120      00:11:19
    172.16.0.2            120      00:12:11
  Distance: (default is 120)
```

Unicast Updates

When configuring Cisco routers for a Frame-Relay network, we may need to configure neighbors in order to send routing updates over the Non-Broadcast Multi-Access network. If you configure frame maps and include the broadcast statement at the end you do not need to configure neighbors! However, if you are prohibited from configuring the broadcast option or if you rely on inverse ARP, then you need to use neighbors on the hub router. Figure 10.1 shows a basic RIP topology and configuration over Frame-Relay. That configuration will NOT work. Also, do not forget you need to disable split horizon on the hub router.

Figure 10.1 *Basic RIP topology and configuration*



```
interface Serial0
 ip address 172.16.0.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.0.5 105
 frame-relay map ip 172.16.0.2 105
 no frame-relay inverse-arp
!
router rip
 network 10.0.0.0
 network 172.16.0.0
```

```
interface Serial0
 ip address 172.16.0.2 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 172.16.0.5 205
 frame-relay map ip 172.16.0.1 205
 no frame-relay inverse-arp
!
router rip
 network 20.0.0.0
 network 172.16.0.0
```

```
interface Serial1.1 multipoint
 ip address 172.16.0.2 255.255.255.0
 no ip split -horizon
 frame-relay map ip 172.16.0.5 205
 frame-relay map ip 172.16.0.1 205
!
router rip
 network 50.0.0.0
 network 172.16.0.0
```

In order to get this topology to work properly, we need to configure neighbors on all three routers.

```
r5(config)# router rip
r5(config-router)# neighbor 172.16.0.1
r5(config-router)# neighbor 172.16.0.2

r1(config)# router rip
r1(config-router)# neighbor 172.16.0.5
r1(config-router)# neighbor 172.16.0.2

r2(config)# router rip
r2(config-router)# neighbor 172.16.0.5
r2(config-router)# neighbor 172.16.0.1
```

RIP Unicast Updates Verification

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set

R     50.0.0.0/8 [120/1] via 172.16.0.5, 00:00:08, Serial0
R     20.0.0.0/8 [120/1] via 172.16.0.2, 00:00:11, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Ethernet0

r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R     50.0.0.0/8 [120/1] via 172.16.0.5, 00:00:11, Serial0
     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, Ethernet0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
R     10.0.0.0/8 [120/1] via 172.16.0.1, 00:00:01, Serial0

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
R     20.0.0.0/8 [120/1] via 172.16.0.2, 00:00:05, Serial1.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial1.1
R     10.0.0.0/8 [120/1] via 172.16.0.1, 00:00:06, Serial1.1
```

OFFSET LIST

The Offset list configuration is the same as IGRP. Remember that RIP uses hop count and a metric of 16 marks a route unreachable. Be careful of how high you set your offset list.

```
r5# show ip route 20.0.0.0
Routing entry for 20.0.0.0/8
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Advertised by rip (self originated)
  Last update from 172.16.0.2 on Serial1.1, 00:00:04 ago
  Routing Descriptor Blocks:
  * 172.16.0.2, from 172.16.0.2, 00:00:04 ago, via Serial1.1
      Route metric is 1, traffic share count is 1

r5(config)# router rip
r5(config-router)# offset-list 0 in 2 serial 1.1

r5# show ip route 20.0.0.0
```

```
Routing entry for 20.0.0.0/8
  Known via "rip", distance 120, metric 3
  Redistributing via rip
  Advertised by rip (self originated)
  Last update from 172.16.0.2 on Serial1.1, 00:00:01 ago
  Routing Descriptor Blocks:
  * 172.16.0.2, from 172.16.0.2, 00:00:01 ago, via Serial1.1
      Route metric is 3, traffic share count is 1
```

SOURCE IP ADDRESS VALIDATION

The Source IP address validation configuration is the same as IGRP.

```
r5(config)# router rip
r5(config-router)# no validate-update-source
```

INTERPACKET DELAY

When a RIP router sends its routing updates, it immediately sends all RIP packets (assuming your RIP update requires more than one packet). A high-end router may overwhelm a low-end router that cannot keep up with the transmission speed. You can configure the high-end router to have a delay between each RIP packet. This delay can be between 8 and 50 milliseconds. Our example puts a 20 millisecond delay in between each RIP update packet.

```
r5(config-router)# output-delay 20
```

TYPICAL GOTCHAS!

- Configuring timers the same on all routers running RIP
- Remembering that RIP V1 is classful and cannot handle discontiguous networks
- Running RIP on non-RIP interfaces

CHAPTER 11

RIP VERSION 2

RIP Version 2 supports authentication, route summarization, and subnet masks. By default, a router running RIP can send and receive both version 1 and 2. But, it only sends version 1. However, both of these defaults can be changed. A router can be restricted to send or receive version 1, version 2, or both.

BASIC RIP VERSION 2 CONFIGURATION

RIP version 2 is configured identical to RIP version 1, except you can change the version globally or on a per-interface basis. Figure 10.1 still applies, except we added the broadcast statement to our frame map so we no longer need that neighbor commands. The example below configures the router to only send and receive version 2 for the entire router.

```
r5(config)# router rip
r5(config-router)# version 2
```

R5 is now sending and receiving version 2 only.

```
r5# show ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 3 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface           Send  Recv   Key-chain
    Ethernet0           2     2
    Serial1.1           2     2
  Routing for Networks:
    50.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.0.1           120      00:04:49
    172.16.0.2           120      00:04:47
  Distance: (default is 120)
```

R1 and R2 can both receive R5's routes, but R5 cannot receive routes from either R1 or R2 because they have not been configured to send version 2. Remember that by default, RIP sends only version 1, but routers can receive either version.

R1 and R2 have not been configured for a specific version. The command below shows the default settings.

```
r1# show ip protocol
Routing Protocol is "rip"
  Sending updates every 30 seconds, next due in 11 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Outgoing update filter list for all interfaces is
  Incoming update filter list for all interfaces is
  Redistributing: rip
  Default version control: send version 1, receive any version
    Interface           Send  Recv  Triggered RIP  Key-chain
    Ethernet0           1     1 2
    Serial0             1     1 2
  Automatic network summarization is in effect
  Routing for Networks:
    10.0.0.0
    172.16.0.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    172.16.0.5           120      00:00:06
    172.16.0.2           120      00:03:12
  Distance: (default is 120)
```

Notice that R1 only receives the route from R5 and not R2. This is because R5 is a hub router. Routes from R2 to R1 and vice versa must go through R5. So, since R5 cannot receive version 1 routes, it cannot send R1 or R2 any routes except its own.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    50.0.0.0/8 [120/1] via 172.16.0.5, 00:00:09, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Ethernet0
```

To correct the problem we just created, we can either configure version 2 globally on R1 and R2 or just configure their respective serial interfaces. Let's assume the lab requires us to run RIP version 1 on both R1 and R2's Ethernet segments.

```
r1(config)# interface s0
r1(config-if)# ip rip send version 2

r2(config)# interface s0
r2(config-if)# ip rip send version 2
```

Verify R1 and R2 both receive each other's routes.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set
```

```
R    50.0.0.0/8 [120/1] via 172.16.0.5, 00:00:07, Serial0
R    20.0.0.0/8 [120/2] via 172.16.0.2, 00:00:07, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R       10.0.0.0/8 [120/4] via 172.16.0.2, 00:00:07, Serial0
C       10.10.10.0/24 is directly connected, Ethernet0

r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    50.0.0.0/8 [120/1] via 172.16.0.5, 00:00:04, Serial0
     20.0.0.0/24 is subnetted, 1 subnets
C       20.20.20.0 is directly connected, Ethernet0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
R    10.0.0.0/8 [120/2] via 172.16.0.1, 00:00:04, Serial0

r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
    default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
R       50.0.0.0/8 [120/2] via 172.16.0.2, 00:00:27, Serial1.1
                   [120/2] via 172.16.0.1, 00:00:12, Serial1.1
C       50.50.50.0/24 is directly connected, Ethernet0
R    20.0.0.0/8 [120/1] via 172.16.0.2, 00:00:27, Serial1.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial1.1
R    10.0.0.0/8 [120/1] via 172.16.0.1, 00:00:12, Serial1.1
```

Did you notice anything strange in the routing table? R5 and R1 both get their own routes back. R2 should get its own route too, but for some reason it did not in our lab. We suspect it could be version dependent since we could find no other explanation why R2 would not receive its own route from R1. This is not a major problem since the connected route is always going to be preferred, but it is recommended to filter your routes whenever you disable split horizon.

```
r5(config)# access-list 50 deny 50.0.0.0
r5(config)# access-list 50 permit any

r5(config)# router rip
r5(config-router)# distribute-list 50 in serial 1.1
```

Now that we've filtered the incoming routes on R5, we no longer get routing table entries for the directly connected Ethernet network

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
```

```
         i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
      default
         U - per-user static route, o - ODR

  Gateway of last resort is not set

       50.0.0.0/24 is subnetted, 1 subnets
  C       50.50.50.0 is directly connected, Ethernet0
  R    20.0.0.0/8 [120/1] via 172.16.0.2, 00:00:27, Serial1.1
       172.16.0.0/24 is subnetted, 1 subnets
  C       172.16.0.0 is directly connected, Serial1.1
  R    10.0.0.0/8 [120/1] via 172.16.0.1, 00:00:19, Serial1.1
```

AUTHENTICATION

RIP authentication has two parts. The first part is key management. The second part is applying authentication to a RIP interface.

KEY MANAGEMENT

Step 1 Identify a key chain. The name of the chain is arbitrary. We chose "RIP" so it is clear what this key chain is being used for.

```
r5(config)# key chain RIP
```

Step 2 Identify the key number. Unless you configure multiple keys, this number should always be "1."

```
r5(config-keychain)# key 1
```

Step 3 Configure the password or "key-string." This is the password that all RIP routers will have to know in order to authenticate.

```
r5(config-keychain-key)# key-string cisco
```

Step 4 (Optional) Specify the time period of how long a key can be received. The first value is start time. The second value can be a specific end time, duration in seconds, or infinite. The default is infinite.

```
r5(config-keychain-key)# accept-lifetime 00:00:00 Nov 1 2001 00:00:00 Nov
   1 2002
```

Step 5 (Optional) Specify the time period of how long a key can be sent. The first value is start time. The second value can be a specific end time, duration in seconds, or infinite. The default is infinite.

```
r5(config-keychain-key)# send-lifetime 00:00:00 Nov 1 2001 00:00:00 Nov 1
   2002
```

Step 6 Verify the key chain configuration using the `show key chain` command.

```
r5# show key chain
Key-chain RIP:
    key 1 -- text "cisco"
        accept lifetime (00:00:00 Nov 1 2001) - (00:00:00 Nov 1 2002)
        send lifetime (00:00:00 Nov 1 2001) - (00:00:00 Nov 1 2001)
```

INTERFACE CONFIGURATION

Step 1 Configure each interface that will use authentication with RIP. Assign the key-chain created in the previous step to the RIP process on that interface.

```
r5(config)# interface serial 1.1
r5(config-if)# ip rip authentication key-chain RIP
```

Step 2 Configure the authentication mode. The mode can be plain text instead of MD5. It is always recommend to use MD5 instead of plain text.

```
r5(config-if)# ip rip authentication mode md5
```

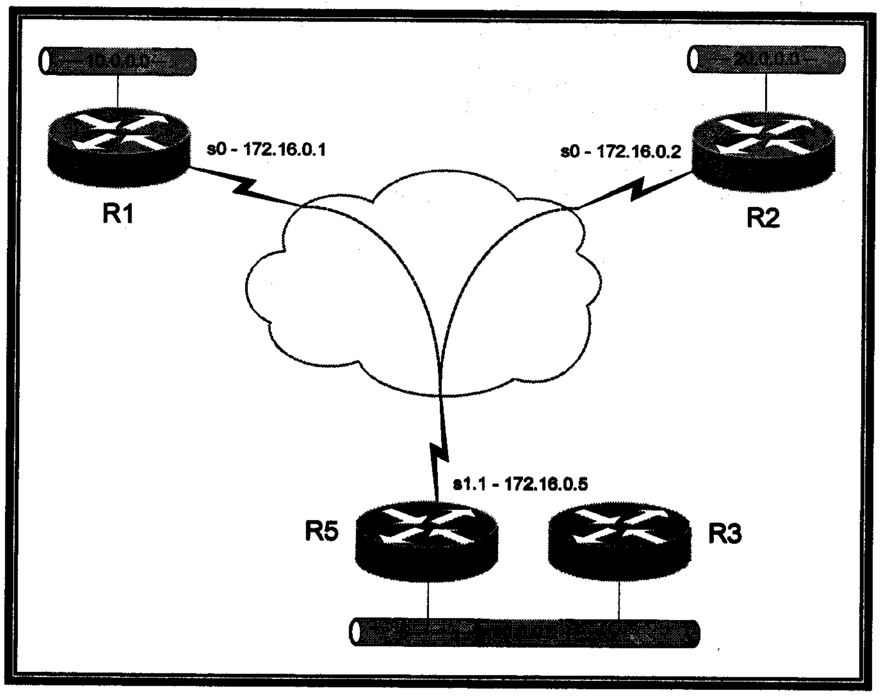
ROUTE SUMMARIZATION

RIP version 2 automatically summarizes routes at the classful boundary. Auto-summary can be disabled using the no auto-summary command. Notice that R1 now sees the 50.0.0.0 route, as a 50.50.50.0 route. R2 should have the same route.

```
r5(config)# router rip
r5(config-router)# no auto-summary

r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
R       50.50.50.0 [120/1] via 172.16.0.5, 00:00:02, Serial0
R    20.0.0.0/8 [120/2] via 172.16.0.2, 00:00:02, Serial0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.0.0 is directly connected, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Ethernet0
```

In figure 11.1, we added an additional router on R5's Ethernet segment. We configured basic RIP version 2 and disabled auto-summary on all routers in figure 11.1

Figure 11.1 *RIP route summarization*



Now all the routes show up on R3 with their actual mask.

```
r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
   area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
R       20.20.20.0 [120/2] via 50.50.50.5, 00:00:24, Ethernet0
     172.16.0.0/24 is subnetted, 1 subnets
R       172.16.0.0 [120/1] via 50.50.50.5, 00:00:24, Ethernet0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.10.10.0 [120/2] via 50.50.50.5, 00:00:24, Ethernet0
```

We decided that we want R5 to advertise R2's network as a summary.

```
r5(config)# interface ethernet 0
r5(config-if)# ip summary-address rip 20.0.0.0 255.0.0.0

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
R       20.0.0.0/8 [120/2] via 50.50.50.5, 00:00:06, Ethernet0
     172.16.0.0/24 is subnetted, 1 subnets
R       172.16.0.0 [120/1] via 50.50.50.5, 00:00:06, Ethernet0
     10.0.0.0/24 is subnetted, 1 subnets
R       10.10.10.0 [120/2] via 50.50.50.5, 00:00:06, Ethernet0
```

Be aware of the limitations of RIP summarization.

- You cannot summarize with a mask smaller than the classful boundary. For example, if summarizing a 10.0.0.0 network, you cannot use a 252.0.0.0 mask. You must at least use a 255.0.0.0 or greater.
- You cannot advertise two networks from the same major network even if your masks are different. This restriction is at the interface level not at the router or network level.
- Split horizon must be disabled (this includes LAN interfaces as well as point-to-multipoint type interfaces).

DEMAND CIRCUIT

In IOS version 12.0(1)T, Cisco introduced a RIP extension to allow for demand circuit operation. You will need a "T" train of code to use this command. With RIP triggered extensions, routing updates are transmitted on the serial interface if one of the following events occurs.

- The router receives a specific request for a routing update. (Full database is sent.)
- Information from another interface causes a change in the routing database. (Only latest changes are sent)
- The interface comes up or goes down. (Partial database is sent.)
- The router is powered on, to ensure that at least one update is sent. (Full database is sent.)

This allows RIP to behave much more like a distance vector protocol. It is currently only available for serial interfaces, but Cisco may decide to extend it to dial circuits such as ISDN thereby enhancing the value of this command on the lab exam. The serial interface must be point-to-point.

```
r1(config-if)# ip rip triggered
RIP: Serial0 is not a point-to-point interface.

r1(config)# interface serial 1.1
r1(config-subif)# ip rip triggered
```

TYPICAL GOTCHAS!

- Remembering to disable off auto-summary
- Having mismatched key numbers or passwords for authentication
- Running RIP on non-RIP interfaces
- Running RIP v2 on interfaces where the proctor expects RIP v1

CHAPTER 12

REDISTRIBUTION

Redistribution is necessary when two different routing protocols need to exchange routes. There are several issues when redistributing routes since the routing protocols typically do not have the same metrics nor do they contain the same type of information. Protocols such as OSPF, send the subnet mask with each route whereas protocols such as RIP do not. There are many issues to consider when redistributing. Some of these include administrative distance, metrics, types of routes that are accepted, and routing loops.

Redistribution Issues

When redistributing OSPF in to BGP, by default, BGP only accepts internal routes not external type 1 or type 2. This may cause a problem if you redistribute other IGP's routes in to OSPF and then on to BGP. Routes redistributed in to OSPF, by default, are external type 2 routes. So, even though these are valid IGP routes, BGP will not accept them by default. The reason for this is to prevent BGP from accepting it's own routes that it advertised to OSPF. It is for loop prevention.

Beware of the metric used by RIP. You will often see configuration examples that use rather large metrics. This is fine for most protocols except RIP. Redistributing in to RIP requires a metric or default-metric. Otherwise, it may get set to 16 and therefore be unreachable.

There are also problems with administrative distance that must be accounted for. For example, IGRP has a better administrative distance than OSPF. Redistribution will often result in OSPF routes being sent to IGRP and then IGRP sending these OSPF routes back as IGRP routes. The router will then accept these new IGRP routes and place them in the routing table. To prevent this problem, as well as others, make sure you filter your redistributed routes.

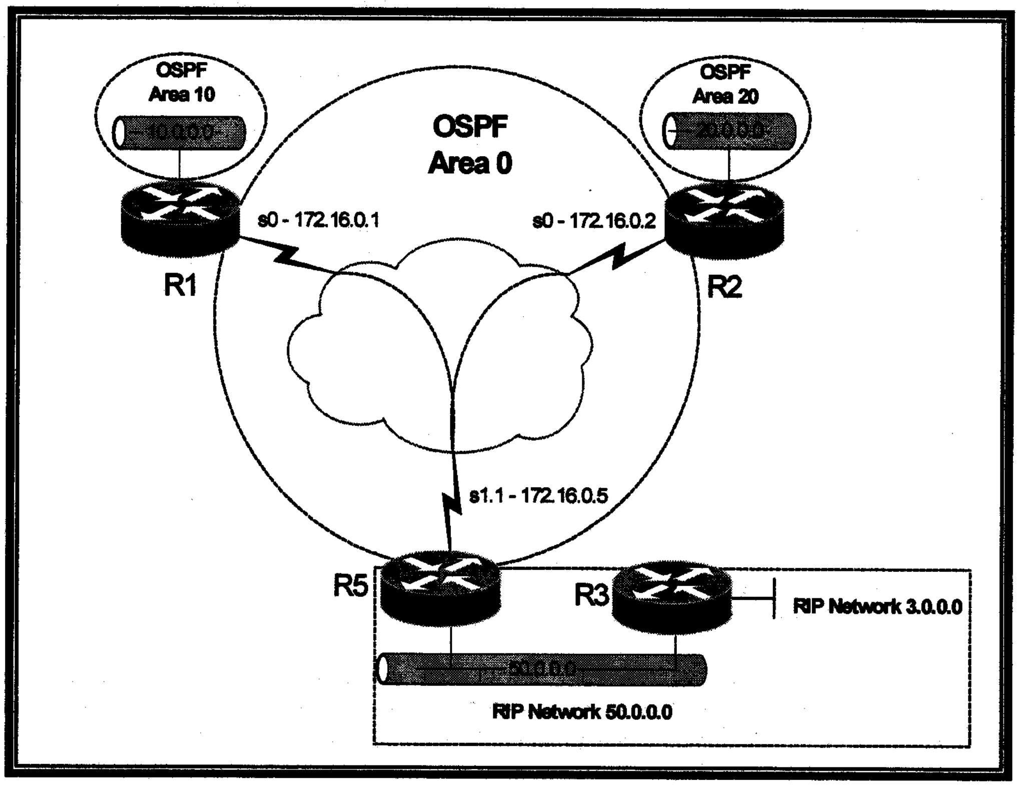
Filtering routes when performing mutual redistribution is very important and without it, your networking routing may not function correctly. Filtering is configured using either route maps or distribute lists. Before you filter your routes, you should configure basic redistribution and confirm that your routing tables are what you expect.

Basic Redistribution

In this example, we configure R5 to redistribute between OSPF and RIP. This particular example does not cause us a direct problem if we do not filter. However, it is always good practice to filter routes whenever configuring mutual redistribution. You may want to check with your lab proctor to see if they expect you to filter mutually redistributed routes regardless of whether it causes a problem or not.

In Figure 12.1, we have a basic OSPF over frame-relay configuration with RIP running on R5 and R3. R3 should receive all OSPF routes and the spoke routers should receive RIP routes after configuring redistribution.

Figure 12.1 *OSPF and RIP*



If using subnetted routes, the "subnet" tag must be configured when redistributing. Unless directed otherwise, always use the subnet option when redistributing in to OSPF.

```
r5(config)# router ospf 1
r5(config-router)# redistribute rip metric 100 subnets
```

When redistributing in to RIP, always assign a metric. Otherwise RIP will automatically assign a metric of 16 to redistributed routes.

```
r5(config)# router rip
r5(config-router)# redistribute ospf 1 metric 2
```

OSPF AND RIP REDISTRIBUTION VERIFICATION

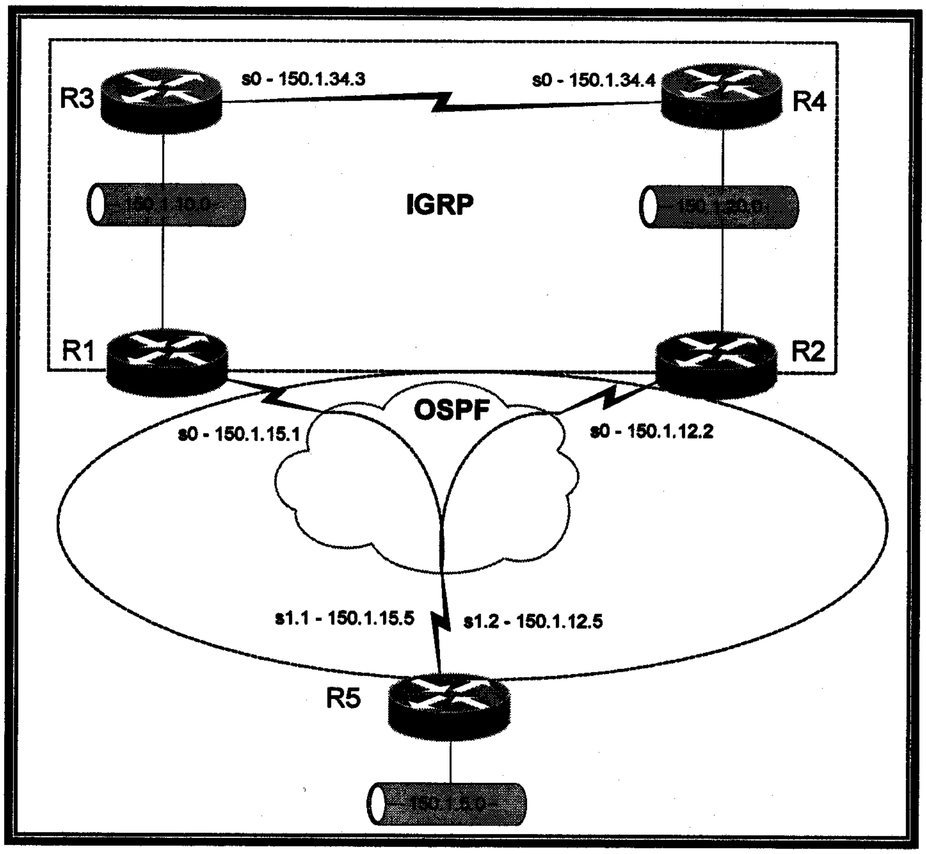
R1 sees the RIP routes as OSPF external type 2 routes. R3 sees the OSPF routes, but on their classful boundary. This is normal behavior and everything works fine with this specific topology.

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
O E2    50.50.50.0 [110/100] via 172.16.0.5, 00:36:13, Serial0
O E2 3.0.0.0/8 [110/100] via 172.16.0.5, 00:36:13, Serial0
     20.0.0.0/24 is subnetted, 1 subnets
O IA    20.20.20.0 [110/138] via 172.16.0.5, 00:36:13, Serial0
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
O       172.16.0.5/32 [110/64] via 172.16.0.5, 00:36:13, Serial0
C       172.16.0.0/24 is directly connected, Serial0
O       172.16.0.2/32 [110/128] via 172.16.0.5, 00:36:13, Serial0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.10.10.0 is directly connected, Ethernet0

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
     3.0.0.0/24 is subnetted, 1 subnets
C       3.3.3.0 is directly connected, Loopback3
R    20.0.0.0/8 [120/2] via 50.50.50.5, 00:00:16, Ethernet0
R    172.16.0.0/16 [120/2] via 50.50.50.5, 00:00:16, Ethernet0
R    10.0.0.0/8 [120/2] via 50.50.50.5, 00:00:16, Ethernet0

r3# ping 20.20.20.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 20.20.20.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/56/60 ms
```

ADMINISTRATIVE DISTANCE ISSUE

Before explaining how to use route filters, lets demonstrate the potential problems if you do not use filters. When redistributing between IGRP and OSPF, you may have a situation where one of your OSPF routers receives an OSPF route originating from IGRP. For example, the OSPF route 150.1.5.0 shows up on R2 as an IGRP route because of redistribution.

Figure 12.2 *IGRP and OSPF Redistribution (Administrative Distance problem)*



Observe Figure 12.2 and follow through as we trace a routing update as it occurred in our test lab.

1. R5 advertises the 150.1.5.0 network to both R1 and R2 over the OSPF frame-relay network
2. R1 and R2 redistribute the 150.1.5.0 network in to IGRP
3. R3 receives the routing update from R1
4. R4 receives the routing update from R2
5. R3 forwards the update to R4
6. R4 does not forward the same update to R3 because of split horizon
7. R4 forwards the update to R2
8. R2 replaces the 150.1.5.0 network in its routing table with the IGRP route because of the administrative distance

OSPF AND RIP ADMINISTRATIVE DISTANCE ISSUE VERIFICATION

```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
```

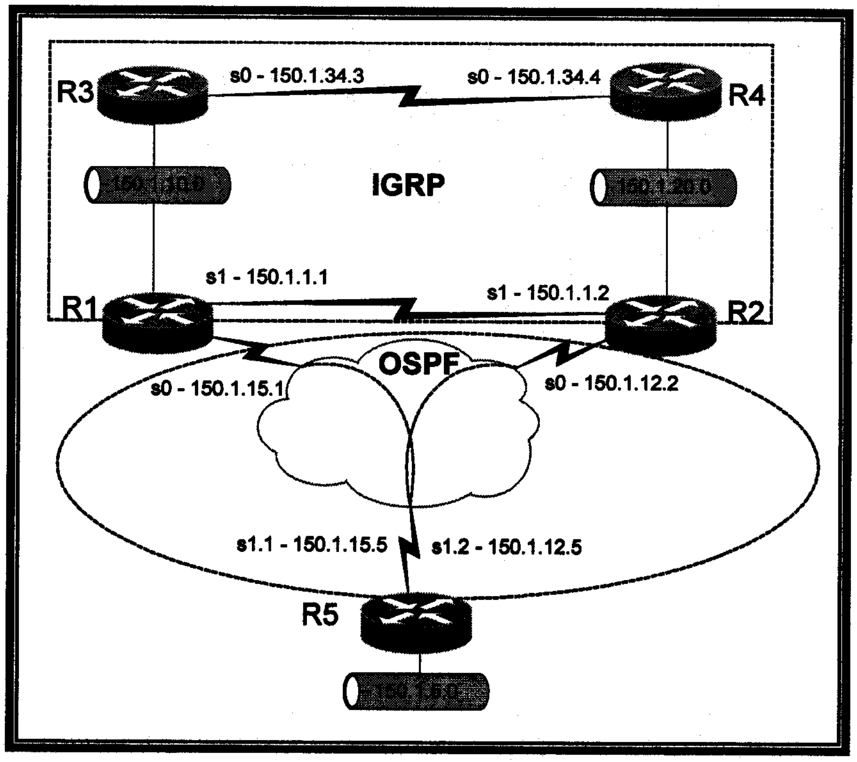
```
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     150.1.0.0/24 is subnetted, 6 subnets
I       150.1.20.0 [100/8676] via 150.1.10.3, 00:00:44, Ethernet0
O       150.1.5.0 [110/74] via 150.1.15.5, 00:02:25, Serial0
C       150.1.15.0 is directly connected, Serial0
I       150.1.12.0 [100/10676] via 150.1.10.3, 00:00:44, Ethernet0
C       150.1.10.0 is directly connected, Ethernet0
I       150.1.34.0 [100/8576] via 150.1.10.3, 00:00:44, Ethernet0

r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     150.1.0.0/24 is subnetted, 6 subnets
C       150.1.20.0 is directly connected, Ethernet0
I       150.1.5.0 [100/8677] via 150.1.20.4, 00:00:30, Ethernet0
I       150.1.15.0 [100/10676] via 150.1.20.4, 00:00:30, Ethernet0
C       150.1.12.0 is directly connected, Serial0
I       150.1.10.0 [100/8676] via 150.1.20.4, 00:00:30, Ethernet0
I       150.1.34.0 [100/8576] via 150.1.20.4, 00:00:30, Ethernet0
```

The result could have been the opposite with R1 having the 150.1.5.0 route installed as an IGRP route instead of R2. Because of split horizon, the routing update is only going to travel in one direction. Note that this results in suboptimal routing. The route is still valid. It just takes a much longer path to get to the destination!

Routing Loop Issue

In the last example, we had a suboptimal path. If we add a link between R1 and R2, we now have a routing loop.

Figure 12.3 *IGRP and OSPF Redistribution (Routing loop problem)*



R1 and R2 advertise the 150.1.5.0 network to R1 as an IGRP route. R1 does not advertise it back to R2 because of split horizon. However, if R1 and R2 transmit simultaneously, you will notice that they temporarily point to each other for that route. Pay close attention to the source address of each route as highlighted below.

```
r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     150.1.0.0/24 is subnetted, 7 subnets
C       150.1.20.0 is directly connected, Ethernet0
I       150.1.5.0/24 is possibly down,
          routing via 150.1.1.1, Serial1
C       150.1.1.0 is directly connected, Serial1
I       150.1.15.0 [100/10476] via 150.1.1.1, 00:01:00, Serial1
C       150.1.12.0 is directly connected, Serial0
I       150.1.10.0 [100/8576] via 150.1.1.1, 00:01:00, Serial1
I       150.1.34.0 [100/8576] via 150.1.20.4, 00:00:16, Ethernet0

r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
```

```
           N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
           E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
           i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
           * - candidate default, U - per-user static route, o - ODR
           P - periodic downloaded static route

Gateway of last resort is not set

     150.1.0.0/24 is subnetted, 7 subnets
I       150.1.20.0 [100/8576] via 150.1.1.2, 00:01:00, Serial1
I       150.1.5.0/24 is possibly down,
          routing via 150.1.1.2, Serial1
C       150.1.1.0 is directly connected, Serial1
C       150.1.15.0 is directly connected, Serial0
I       150.1.12.0 [100/10476] via 150.1.1.2, 00:01:00, Serial1
C       150.1.10.0 is directly connected, Ethernet0
I       150.1.34.0 [100/8576] via 150.1.10.3, 00:00:09, Ethernet0
```

After R1 stops sending the route to R2 because of split horizon, R2 will only receive the route from R4.

```
r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     150.1.0.0/24 is subnetted, 7 subnets
C       150.1.20.0 is directly connected, Ethernet0
I       150.1.5.0/24 is possibly down,
          routing via 150.1.20.4, Ethernet0
C       150.1.1.0 is directly connected, Serial1
I       150.1.15.0 [100/10476] via 150.1.1.1, 00:00:16, Serial1
C       150.1.12.0 is directly connected, Serial0
I       150.1.10.0 [100/8576] via 150.1.1.1, 00:00:16, Serial1
I       150.1.34.0 [100/8576] via 150.1.20.4, 00:00:18, Ethernet0
```

ROUTE MAPS

Route maps allow for very granular control over the redistribution process. Route maps allow you to specify an access-list to permit or deny routes from being redistributed, assign metrics, and tag routes. A route map essentially has two important functions: match and set. The highlighted options below show the most useful commands when redistributing with route maps.

```
r5(config)# route-map OSPF_TO_RIP permit 10
r5(config-route-map)# match ?
  as-path      Match BGP AS path list
  clns         CLNS information
  community    Match BGP community list
  interface    Match first hop interface of route
  ip           IP specific information
  length       Packet length
  metric       Match metric of route
  route-type   Match route-type of route
  tag          Match tag of route

r5(config-route-map)# set ?
  as-path            Prepend string for a BGP AS-path attribute
  automatic-tag      Automatically compute TAG value
```

```
clns                OSI summary address
comm-list           set BGP community list (for deletion)
community           BGP community attribute
dampening           Set BGP route flap dampening parameters
default             Set default information
interface           Output interface
ip                  IP specific information
level               Where to import route
local-preference    BGP local preference path attribute
metric              Metric value for destination routing protocol
metric-type         Type of metric for destination routing protocol
origin              BGP origin code
tag                 Tag value for destination routing protocol
weight              BGP weight for routing table
```

The administrative distance and routing loop problem can both be resolved by filtering with route maps.
To fix this problem we need to filter the routes between the two protocols. There are four steps to this process.

Step 1 Identify the OSPF and IGRP routes. You should do this BEFORE configuring any redistribution so it's clear which routes belong to which protocol.

```
r5# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate
default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     50.0.0.0/24 is subnetted, 1 subnets
C       50.50.50.0 is directly connected, Ethernet0
I    3.0.0.0/8 [100/1600] via 50.50.50.3, 00:00:09, Ethernet0
     20.0.0.0/24 is subnetted, 1 subnets
O IA    20.20.20.0 [110/74] via 172.16.0.2, 00:31:38, Serial1.1
     172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks
C       172.16.0.0/24 is directly connected, Serial1.1
O       172.16.0.1/32 [110/64] via 172.16.0.1, 00:31:39, Serial1.1
O       172.16.0.2/32 [110/64] via 172.16.0.2, 00:31:39, Serial1.1
O IA 10.0.0.0/8 [110/74] via 172.16.0.1, 00:31:39, Serial1.1
```

Step 2 Configure access-lists to be used for OSPF and IGRP. We do not need to configure the host routes from the frame-relay interfaces since the 172.16.0.0 /24 will be the only route actually advertised to IGRP. If we did not have a route to the 172.16.0.0 network, then we would enter them in the access-list. But, that would not be possible in this situation since that network is directly connected.

```
r5(config)# access-list 10 permit 10.0.0.0 0.255.255.255
r5(config)# access-list 10 permit 20.20.20.0 0.0.0.255
r5(config)# access-list 10 permit 172.16.0.0 0.0.0.255

r5(config)# access-list 20 permit 3.0.0.0 0.255.255.255
```

Step 3 Configure your route maps to be used by IGRP. Route maps use sequence numbers. This determines the order the route map will be processed. The second route map sequence in both examples says to match all traffic and deny it. Since the "permit 10" sequence is processed first, it already matches the desired routes and lets them pass. If a route is not matched by the first sequence, it is denied by the second sequence.

Do not forget to either set the metric at the route map level or the redistribution level. In the previous example, we configured the metric with the redistribution command.

```
r5(config)# route-map OSPF_TO_IGRP permit 10
r5(config-route-map)# match ip address 10
r5(config-route-map)# set metric 50 1 255 1 1500

r5(config-route-map)# route-map OSPF_TO_IGRP deny 20
```

Step 4 Configure your route maps to be used by OSPF.

```
r5(config-route-map)# route-map IGRP_TO_OSPF permit 10
r5(config-route-map)# match ip address 20
r5(config-route-map)# set metric 100

r5(config-route-map)# route-map IGRP_TO_OSPF deny 20
```

Step 5 Configure your redistribution and apply the route map you just created.

```
r5(config-route-map)# router ospf 1
r5(config-router)# redistribute igrp 1 subnets route-map IGRP_TO_OSPF

r5(config-router)# router igrp 1
r5(config-router)# redistribute ospf 1 route-map OSPF_TO_IGRP
```

Distribute Lists

Distribute lists can solve the same problem as route maps, but they are less granular. You can only permit or deny routes. You cannot set metrics or tags. Multiple distribute-lists can be used on a routing process if combining an interface and routing protocol distribute list. For example, you may configure a router to never accept any RFC 1918 routes at the routing protocol level. Then, you filter specific routes at the interface level. Note that the routes must be allowed by BOTH distribute-lists to be placed in the routing table.

The administrative distance and routing loop problem can both be solved using distribute lists. Like route maps, the first step is to identify which routes belong to each protocol. We can see that OSPF only has two routes: 150.1.5.0/24 and 150.1.15.0/24. IGRP routes include 150.1.1.0/24, 150.1.10.0/24, 150.1.20.0/24, and 150.1.34.0/24.

Step 1 Configure the access-lists that will be used by OSPF and IGRP to match traffic.

```
r1(config)# access-list 10 permit 150.1.5.0 0.0.0.255
r1(config)# access-list 10 permit 150.1.15.0 0.0.0.255

r1(config)# access-list 20 permit 150.1.1.0 0.0.0.255
r1(config)# access-list 20 permit 150.1.10.0 0.0.0.255
r1(config)# access-list 20 permit 150.1.20.0 0.0.0.255
r1(config)# access-list 20 permit 150.1.34.0 0.0.0.255

r2(config)# access-list 10 permit 150.1.5.0 0.0.0.255
r2(config)# access-list 10 permit 150.1.12.0 0.0.0.255

r2(config)# access-list 20 permit 150.1.1.0 0.0.0.255
r2(config)# access-list 20 permit 150.1.10.0 0.0.0.255
r2(config)# access-list 20 permit 150.1.20.0 0.0.0.255
r2(config)# access-list 20 permit 150.1.34.0 0.0.0.255
```

Step 2 Apply the access-lists to the distribute list. You must redistribute outbound routes and not inbound routes. Cisco routers only allow you to filter inbound routes globally or on an interface level. We want to filter routes from a specific routing protocol which you cannot do with the "in" option. Notice the different options available when filtering inbound or outbound.

```
r1(config-router)# distribute-list 10 in ?
  Ethernet   IEEE 802.3
  Loopback   Loopback interface
  Null       Null interface
  Serial     Serial
  <cr>

r1(config-router)# distribute-list 10 out ?
  Ethernet    IEEE 802.3
  Loopback    Loopback interface
  Null        Null interface
  Serial      Serial
  bgp         Border Gateway Protocol (BGP)
  connected   Connected
  egp         Exterior Gateway Protocol (EGP)
  eigrp       Enhanced Interior Gateway Routing Protocol (EIGRP)
  igrp        Interior Gateway Routing Protocol (IGRP)
  ospf        Open Shortest Path First (OSPF)
  rip         Routing Information Protocol (RIP)
  static      Static routes
  <cr>

r1(config)# router ospf 1
r1(config-router)# distribute-list 20 out igrp 1

r1(config-router)# router igrp 1
r1(config-router)# distribute-list 10 out ospf 1

r2(config-router)# router ospf 1
r2(config-router)# distribute-list 20 out igrp 1

r2(config-router)# router igrp 1
r2(config-router)# distribute-list 10 out ospf 1
```

OSPF to RIP Distribute Lists Verification

Our routing now looks correct on both R1 and R2. The 150.1.5.0 network now shows up as an OSPF route.

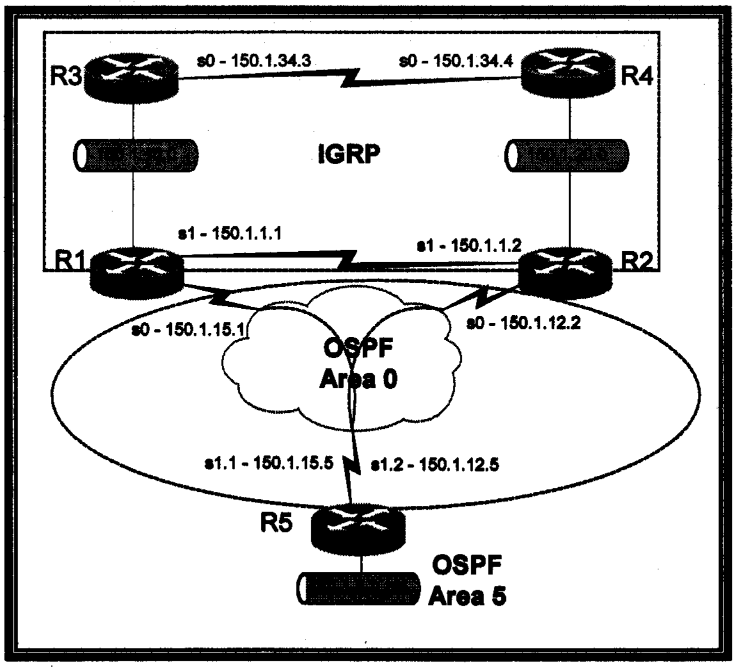
```
r1# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected, Loopback0
     150.1.0.0/24 is subnetted, 7 subnets
I       150.1.20.0 [100/8576] via 150.1.1.2, 00:00:12, Serial1
O       150.1.5.0 [110/74] via 150.1.15.5, 00:00:49, Serial0
C       150.1.1.0 is directly connected, Serial1
C       150.1.15.0 is directly connected, Serial0
I       150.1.12.0 [100/10476] via 150.1.1.2, 00:00:12, Serial1
C       150.1.10.0 is directly connected, Ethernet0
I       150.1.34.0 [100/8576] via 150.1.10.3, 00:00:22, Ethernet0
```

```
r2# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/24 is subnetted, 1 subnets
C       2.2.2.0 is directly connected, Loopback0
     150.1.0.0/24 is subnetted, 7 subnets
C       150.1.20.0 is directly connected, Ethernet0
O       150.1.5.0 [110/74] via 150.1.12.5, 00:00:19, Serial0
C       150.1.1.0 is directly connected, Serial1
I       150.1.15.0 [100/10476] via 150.1.1.1, 00:00:19, Serial1
C       150.1.12.0 is directly connected, Serial0
I       150.1.10.0 [100/8576] via 150.1.1.1, 00:00:19, Serial1
I       150.1.34.0 [100/8576] via 150.1.20.4, 00:00:20, Ethernet0
```

VLSM TO FLSM ISSUE

So far, all of our redistribution examples had subnets on classful boundaries. How does IGRP and RIP handle a redistributed network with a subnet mask on a non-classful boundary. It does not! We need to make sure we summarize or aggregate all of our routes prior to sending them to FLSM routing protocols. OSPF, EIGRP, IS-IS, and BGP all support variable length subnets masks. RIP and IGRP do not support variable length subnet masks. They only support fixed length subnet masks (FLSM).

Figure 12.4 *Redistribution from VLSM to FLSM*



```
r5(config)# router ospf 1
r5(config-router)# area 5 range 150.1.5.0 255.255.255.0
```

FLSM TO VLSM REDISTRIBUTION VERIFICATION

We check our routing tables again and now R3 can reach the 150.1.5.0 network.

```
r3# show ip route 150.1.5.5
Routing entry for 150.1.5.0/24
  Known via "igrp 1", distance 100, metric 6536
  Redistributing via igrp 1
  Advertised by igrp 1 (self originated)
  Last update from 150.1.10.1 on Ethernet0, 00:00:16 ago
  Routing Descriptor Blocks:
  * 150.1.10.1, from 150.1.10.1, 00:00:16 ago, via Ethernet0
      Route metric is 6536, traffic share count is 1
      Total delay is 1010 microseconds, minimum bandwidth is 1554 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 0

r3# ping 150.1.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
```

What would we do if area 0 was changed to a /28 mask as shown in Figure 12.4? We know that the area range command won't work. So, the only other option would be to create a static route using the classful range and redistribute that static in to IGRP. The problem with this option is that typically we are not allowed to add any static routes. Should you come across this situation, ask the proctor if you may configure a static route because there is no other way to summarize the directly connected network.

TYPICAL GOTCHAS!

- Not filtering routes during redistribution
- Not specifying a metric for routes redistributed into RIP or EIGRP
- Missing the 'subnet' command for routes redistributed into OSPF

Section III

Cisco General Security and Firewalls

CHAPTER 13

CISCO ROUTER SECURITY

CISCO ROUTER SECURITY RECOMMENDATIONS

- Disable unnecessary services
- Denial of Service prevention
- Rate limiting traffic
- Router self protection and crash dumps
- Black Hole routes

DISABLE UNNECESSARY SERVICES

As with any networked device, unnecessary or unused services should always be disabled. The more services your device provides, the more potential security holes. Cisco routers have a unique set of security concerns because of their function of routing traffic. By default, many services are disabled depending on the IOS version. However, it is best to keep a checklist of services that should be disabled unless necessary. Below is a basic list that we compiled.

- CDP
- Diagnostic ports (TCP and UDP small-servers)
- HTTP server
- Finger
- DHCP and BOOTP server
- ICMP unreachable
- ICMP redirects
- Proxy ARP

CISCO DISCOVERY PROTOCOL (CDP)

CDP provides information about directly Cisco devices. It operates at layer 2. CDP can provide potentially useful information for an attacker such as IP address, software version, platform (hardware), and capabilities (such as whether the device is a router, switch, bridge, etc.). It should always be disabled on public links at a minimum and throughout your network depending on your security policy. CDP can be disabled globally or on a per-interface basis.

```
r5# show cdp neighbor detail
-------------------------
Device ID: 006778391
Entry address(es):
  IP address: 10.10.199.100
Platform: WS-C2926,  Capabilities: Trans-Bridge Source-Route-Bridge Switch
Interface: Ethernet0,  Port ID (outgoing port): 2/5
Holdtime : 171 sec
Version :
WS-C2926 Software, Version McpSW: 4.5(10) NmpSW: 4.5(10)
Copyright (c) 1995-2000 by Cisco Systems
```

To disable CDP globally, enter the **no cdp run** command in global configuration mode.

```
r5(config)# no cdp run
```

Alternatively, if CDP is needed on some links and you want to disable it on public links, enter the **no cdp enable** command in interface configuration mode.

```
r5(config)# interface ethernet 0
r5(config-if)# no cdp enable

r5# show cdp neighbors
% CDP is not enabled
```

Diagnostic Ports

An attacker can send a large number of packets attempting to attach to these ports using random, spoofed addresses. The router can become overwhelmed trying to respond to these illegitimate requests thereby causing a denial of service. It is highly advised that this service always be disabled. By default, they are disabled in IOS version 12.0 and later.

The following list shows the diagnostic ports available for both TCP and UDP. These are the ports that an attacker will try to connect to.

TCP small servers

- Echo
- Chargen
- Discard
- Daytime

UDP small servers

- Echo
- Discard
- Chargen

These services only need to be disabled in versions prior to 12.0 or if another administrator enabled them.

```
r5(config)# no service tcp-small-servers
r5(config)# no service udp-small-servers
```

HTTP Server

The HTTP server allows for configuration and management of the device from a web browser. There are two security advisories pertaining to HTTP server on a Cisco device. The first vulnerability is IOS HTTP Authorization. This vulnerability allows an attacker to execute commands at the highest privilege level without passing through proper authentication. The HTTP server service and local authentication must be configured for an attacker to exploit this vulnerability. An attacker simply needs to enter the URL in the following format and try up to 84 different combinations (numbers between 16 and 99 in place of XX).

```
http://<device_ip_address>/level/XX/exec/....
```

This vulnerability is documented as Cisco Bug ID CSCdt93862. Check if your software version is subject to this vulnerability.

THERE ARE THREE REMEDIES TO THIS SECURITY HOLE.

1. Disable the HTTP server service
2. Use TACACS or RADIUS instead of local authentication for your HTTP authentication
3. Limit HTTP access to only authorized IP addresses or networks using an access-class (the router will still be subject to attack from these authorized IP's)

The second HTTP vulnerability allows an attacker to reload the router thereby resulting in a denial of service. An attacker can browse to http://<router_ip_address>/%% and cause the router to crash and reload. This is due to the router incorrectly parsing the "%%" thereby entering an infinite loop. A watchdog timer monitoring for this event forces the router to reboot two minutes after this loop condition is detected in order to restore service. However, an attacker can do this every time the router comes up. Sometimes, the router has difficultly reloading and must be manually power cycled.

THERE ARE THREE REMEDIES TO THIS SECURITY HOLE.

1. Disable the HTTP server service
2. Upgrade to a fixed software version
3. Limit HTTP access to only authorized IP addresses or networks using an access-class (the router will still be subject to attack from these authorized IP's)

FINGER

Finger is used to find out what users that are logged in to a system as well as other information including the processes running on the system, the line number, connection name, idle time, and terminal location. Always disable this service unless it is required. At a minimum, do not allow Finger from the Internet or other public links by filtering at the entrance points of your network. It can only be disabled globally.

```
r5(config)# no ip finger
```

DHCP AND BOOTP SERVER

DHCP and BootP are used to allocate IP addresses and other IP related information to clients. These services are typically not exploited if they are enabled, but it is always advised to disable any service not in use. A DHCP server must be explicitly defined before it can be disabled. So, if the router has never been configured for DHCP then you do not need to disable it. BootP on the other hand, can be disabled using the **no ip bootp server** command.

ICMP UNREACHABLES

If a Cisco router receives a nonbroadcast packet that uses an unknown protocol, the router sends an "ICMP Protocol Unreachable" message back to the source. If a Cisco router receives a packet that it is unable to deliver to the destination because it does not have a route to the destination address, it sends an "ICMP Host Unreachable" message to the source. This can provide an attacker information about the protocols a host understands or the networks a router knows how to reach.

There is a specific security advisory for the GSR 12000 series routers that can force a denial of service if the router is sending too many ICMP unreachables typically during a period of heavy network scanning.

ICMP unreachables are enabled by default and should be disabled on all interfaces unless they are needed. It can be disabled at the interface level only.

```
r5(config)# interface ethernet 0
r5(config-if)# no ip unreachables
```

If unnreachables cannot be disabled, the router can be configured to rate-limit the amount of ICMP unreachables sent. By default, a Cisco router only sends one packet per 500 ms to protect against denial of service attacks. This feature became available in version 12.1. The rate limit is for the aggregate of all ICMP unreachables as shown in the list below.

ICMP type 3 codes:
0 = net unreachable
1 = host unreachable
2 = protocol unreachable
3 = port unreachable
4 = fragmentation needed and DF set
5 = source route failed

This rate-limit does not affect other packets like ICMP echo requests (ping) or ICMP "time exceeded" (traceroute) messages. The command below limits a router to sending only one unreachable message every 1000ms.

```
r5(config)# ip icmp rate-limit unreachable 1000
```

Configuring the rate-limit for code 4 (fragmentation and DF set) requires a separate command as shown below.

```
r5(config)# ip icmp rate-limit unreachable DF 1000
```

ICMP Redirect Messages

If a router has to send traffic out the same interface it was received, it will send an ICMP Redirect message to the source host to let it know to forward those packets to the other router next time. This feature is enabled by default except when Hot Standby Router Protocol (HSRP) is configured on an interface. Redirects are disabled at the interface level.

```
r5(config)# interface ethernet 0
r5(config-if)# no ip redirects
```

Cisco routers send ICMP redirects when ALL of the following conditions are met:

1. The interface on which the packet comes into the router is the same interface that the packet gets routed out
2. The network of the source IP address is the same network of the next hop IP address of the packet
3. The datagram is not source-routed
4. The router is configured to send redirects (enabled by default)

Proxy Address Resolution Protocol (ARP)

Proxy ARP can help hosts on a subnet reach remote subnets without configuring routing or a default gateway. The router will reply to the proxy ARP request with its MAC address. Hosts will then send all traffic for this destination address to the router, and the router will forward the traffic based on its routing table. This makes spoofing much easier and can reveal internal addresses if NAT is not being used. Proxy ARP should never be used. Disable Proxy ARP at the interface level.

```
r5(config)# interface ethernet 0
r5(config-if)# no ip proxy-arp
```

Preventing most Denial of Service attacks

There really is no way to completely prevent DoS attacks. However, there are ways to mitigate its effects and in prevent them in some cases. The first thing you should do on all border routers is apply ingress and egress route filters. These filters should also include blocking RFC 1918 addresses as shown below.

- 10.0.0.0
- 192.168.0.0
- 172.16.0.0 – 172.16.31.0

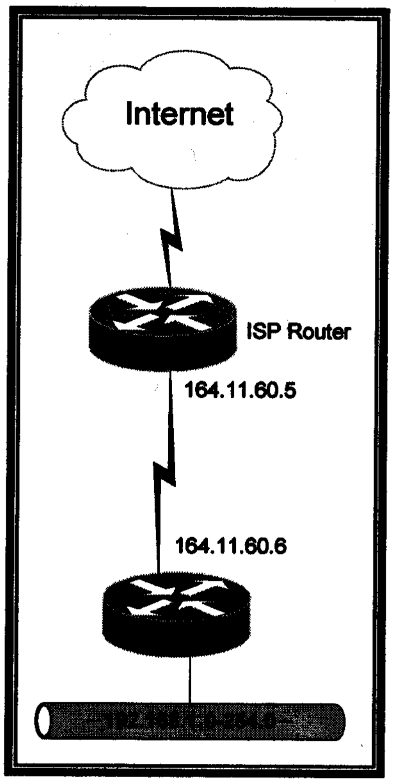
Other features such as CEF with Unicast Reverse Path checking, disabling redirected broadcasts, TCP intercept, and Committed Access Rate can all be used to help against DoS attacks.

Ingress and Egress filtering

Ingress filtering is blocking traffic on an outside interface that has a source address of your inside interface(s). Typically, we perform this sort of filtering on the exit points of your network.

Egress filtering is the opposite of ingress filtering. Frequently, when an attacker compromises a host that he/she will use as an attacker, they will spoof the source address. If they compromise one of your hosts, you want to make sure that if they try to spoof the source address that you do not permit the firewall or router to allow this traffic out of the network.

Figure 13.1 *Ingress and egress filtering*



In Figure 13.1, we see a simple Internet connection. Based on this topology, we want to block all incoming packets with a source from RFC 1918 addresses, internal networks, loopback addresses, multicast, and our serial interface IP address as well as only permit our internal networks from originating outbound connections. Make sure you do not need to permit any of these addresses for any reason. You may need to allow multicast packets if you are running a routing protocol such as OSPF that uses multicast. Pay close attention to the subnet masks we are using.

```
r13(config)# access-list 100 deny ip 127.0.0.0 0.255.255.255 any log-input
r13(config)# access-list 100 deny ip 10.0.0.0 0.255.255.255 any log-input
r13(config)# access-list 100 deny ip 172.16.0.0 0.15.255.255 any log-input
r13(config)# access-list 100 deny ip 192.168.0.0 0.0.255.255 any log-input
r13(config)# access-list 100 deny ip 224.0.0.0 31.255.255.255 any log-input
r13(config)# access-list 100 deny ip host 164.11.60.6 any log-input
r13(config)# access-list 100 permit ip any any
r13(config)# interface serial 0/0
r13(config-if)# ip access-group 100 in

r13(config)# access-list 101 permit ip 192.168.0.0 0.0.255.255 any log-input
r13(config)# interface ethernet 0
r13(config-if)# ip access-group 101 in
```

Cisco Express Forwarding (CEF) and Unicast Reverse Path (RPF)

CEF enhances performance and security by using an optimal routing structure that allows it to perform well when routing heavy traffic to a high number of destinations. CEF with Unicast RPF helps prevent spoofing attacks by verifying the source of the packet. It does this by checking to make sure there is a valid path to the source via the interface in which it arrived. CEF checks the routing table for a valid path. One of the requirements to run CEF with Unicast RPF is that your routing must be symmetric. In other words, if a packet takes a certain path to arrive at its destination, the reply packet cannot take a different path. Otherwise, CEF cannot verify that the source is actually valid. Prior to configuring CEF, ensure your hardware and software are compatible. In the lab exam, this should already be done for you. They cannot expect you to configure a feature that is not supported!

Step 1 Enable CEF on a global basis with the **ip cef** command.

```
r13(config)# ip cef
```

Step 2 Configure the specific interfaces for the RPF check. Depending on your IOS version, you may also have an access-list option for your RPF check. If you do not add an access-list, all traffic without a reverse path is immediately dropped. The access-list option can be used to integrate your ingress access-lists with RPF. In our example, we are using access-list 100 that we created in the previous section.

```
r13(config)# interface serial 0/0
r13(config-if)# ip verify unicast reverse-path 100
```

CEF VERIFICATION

There are several show commands that are useful when checking if CEF is configured and how well it is performing. The **show ip traffic** command was edited for brevity.

```
r13# show cef interface serial0/0
Serial0/0 is up (if_number 3)
  Corresponding hwidb fast_if_number 3
  Corresponding hwidb firstsw->if_number 3
  Internet address is 164.11.60.6/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is enabled
  Inbound access list is not set
  Outbound access list is not set
  IP policy routing is disabled
  Interface is marked as point to point interface
  Hardware idb is Serial0/0
  Fast switching type 4, interface type 56
  IP CEF switching enabled
  IP CEF Feature Fast switching turbo vector
  Input fast flags 0x4000, Output fast flags 0x0
  ifindex 2(2)
  Slot 0 Slot unit 0 VC -1
  Transmit limit accumulator 0x0 (0x0)
  IP MTU 1500

r13# show ip traffic
IP statistics:
  Rcvd:  1575380 total, 37494 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other
  Frags: 0 reassembled, 0 timeouts, 0 could not reassemble
         3 fragmented, 0 could not fragment
  Bcast: 870 received, 0 sent
  Mcast: 0 received, 0 sent
  Sent:  3664 generated, 796778 forwarded
```

```
Drop:  971 encapsulation failed, 0 unresolved, 0 no adjacency
       0 no route, 0 unicast RPF, 0 forced drop
```

TCP SYN ATTACKS

A TCP SYN attack occurs when an attacking host(s) attempts to connect to one of your internal servers, typically a web server, and is not available (usually because the source address is spoofed) to complete the connection request. Your server now has a half-open connection. Eventually, your server will have too many half open connections and will run out of resources to respond to new requests .

The impact of a TCP SYN attack can be limited using three methods. The preferred method is to let the hosts determine when to terminate half open connections. It is difficult for network equipment to know when a server is in danger of running low on resources. However, Cisco routers can be configured to intervene and prevent these attacks using TCP Intercept or TCP Watch. The last option is to rate limit TCP SYN connections with Committed Access Rate. The latter two options are discussed in the upcoming sections.

TCP INTERCEPT AND WATCH

In order to combat TCP SYN attacks, Cisco routers can be configured to intervene on a hosts behalf. TCP Intercept captures packets destined for a host and responds on behalf of the destination host. If the source responds, the router allows the connection to be completed from the source to the destination. TCP Watch tracks the connection requests and will send a message to the destination host and instruct it to terminate half-opens based on predefined thresholds.

There are two issues involving TCP Intercept. First, do not configure TCP Intercept with Black Hole routes. Second, make sure your firewall will issue a TCP Reset (RST) for denied connections

COMMITTED ACCESS RATE (CAR)

CAR can be used to limit the amount of bandwidth for specific traffic types. Additionally, it can even help limit SYN connections thereby limiting the impact of a TCP SYN attack. Prior to configuring CAR, make sure you are running CEF. The interface that will use CAR must be a CEF switched interface as shown in the example below.

```
r13# show cef interface serial 0/0
Serial0/0 is up (if_number 3)
  Corresponding hwidb fast_if_number 3
  Corresponding hwidb firstsw->if_number 3
  Internet address is 164.11.60.6/24
  ICMP redirects are always sent
  Per packet load-sharing is disabled
  IP unicast RPF check is enabled
  Inbound access list is 100
  Outbound access list is 101
  IP policy routing is disabled
  Interface is marked as point to point interface
  Hardware idb is Serial0/0
  Fast switching type 4, interface type 56
  IP CEF switching enabled
  IP CEF Feature Fast switching turbo vector
  Input fast flags 0x4005, Output fast flags 0x1
  ifindex 2(2)
  Slot 0 Slot unit 0 VC -1
```

```
    Transmit limit accumulator 0x0 (0x0)
    IP MTU 1500
```

The most difficult part about configuring CAR is determining what limits you want to place on the traffic being rate limited. In the lab exam, if they do not indicate the values (or percentage of the bandwidth) then you should be free to select any values that you feel are reasonable. In a production network however, you have to analyze your traffic patterns very carefully as you do not want to prevent legitimate traffic from being denied.

After you have verified that you are running CEF on the interface, you'll need to identify which traffic you want to rate limit. In our first example, we are concerned about limiting the potential damage caused by attacks that use the ICMP echo and echo-reply protocol.

```
r13(config)# access-list 150 permit icmp any any echo
r13(config)# access-list 150 permit icmp any any echo-reply
```

Then we create the rate-limit and assign the limits. Typically, we will want to rate limit incoming traffic on the customer end as well as outbound traffic towards customers at the service provider edge. After the access-list number (150), there are three values. The first value (8000) is the bandwidth limit in bits per second. The second value (2000) is the normal burst in bytes. The third value (2000) is the maximum burst in bytes. As long as the traffic ICMP traffic is not consuming more than 8000 bps, the router will continue to let the traffic through. If it exceeds 8000 bps, the traffic is dropped.

```
r13(config)# interface serial 0/0
r13(config-if)# rate-limit input access-group 150 8000 2000 2000 conform-
   action transmit exceed-action drop
```

Our second example illustrates how to limit the TCP SYN connections using CAR. The first line of the access-list has a "deny" which means do not subject to CAR. Established TCP sessions are obviously not considered a SYN attack since the very nature of a TCP SYN attack is that these TCP sessions are never established.

```
r13(config)# access-list 170 deny tcp any any established
r13(config)# access-list 170 permit tcp any any
```

We had to increase our bandwidth limitation from the previous example because TCP packets are much more frequent than ICMP echo or echo-reply packets. Once again, make sure that these values are appropriate for your network

```
r13(config)# interface serial 0/0
r13(config-if)# rate-limit input access-group 170 64000 2000 2000 conform-
   action transmit exceed-action drop
```

PING OF DEATH

A Ping of Death is when an attacker sends a large ICMP packet (greater than 65535 bytes) to a target host. Since 65535 bytes is too large for a single IP packet, it is fragmented in to many smaller packets. As the target is attempting to reassemble the packet, the buffer overflows and can cause the target to reboot or hang. Fortunately, most modern operating systems do not have trouble with these invalid packet types. Check your operating system documentation and release notes to see if your hosts are vulnerable. ICMP, specifically ICMP fragments, can be blocked at your network entrance points to provide some protection in case your hosts cannot be immediately upgraded or patched.

The example below shows a partial access-list that is applied to a serial interface. Based on this configuration, all traffic will be denied so make sure you permit some traffic (assuming you are not running a feature set such as CBAC that will dynamically open holes in the access list as needed).

```
r13(config)# access-list 100 deny icmp any any fragments
r13(config)# interface serial 0/0
r13(config-if)# ip access-group 100 in
```

SMURF ATTACKS

Smurf attacks occur when an attacker sends a large number of ICMP packets to a broadcast. The attacker uses a spoofed source address. So, the first step to combat Smurf attacks is to configure anti-spoofing using ingress and egress access-lists and/or CEF with Unicast RPF. The second step is to disable IP directed broadcasts and all interfaces.

```
r13(config)# interface serial 0/0
r13(config-if)# no ip directed-broadcast
```

ROUTER SELF PROTECTION

If an attacker targets the router itself instead of hosts, the router must be able to continue to function. The router can protect itself from crashing using CEF and by configuring it to allow some resources for normal router operation. CEF allows the router to handle high loads of traffic without too much difficulty when other switching methods would probably crash.

Scheduler allocate or interval can be configured to ensure the router is not overburdened fast switching packets and still has interrupts to attend to router functions. This prevents an attacker from overwhelming a router with traffic (typically bogus traffic) causing it to run out of resources for normal router operations and consequently crashing.

To configure scheduler allocate, you can choose to accept the default values (varies by IOS version) by simply entering the **scheduler allocate** global command. Alternatively, you can specifically specify the amount of time in microseconds spent on interrupts or processes. The first value in the scheduler allocate is how much time the router will spend on fast switching. The second value is how much time is the router is guaranteed to handle process level functions such as handling routing protocol updates.

```
r13(config)# scheduler allocate 4000 1000
```

Based on the values shown, this would allow the router to spend 25% of the time on process level functions. To calculate the percentage, use the following formula.

[process time] \ [interrupt time]

[1000] \ [4000]
1000 \ 4000 = .25 or 25%

Scheduler interval controls the amount of time that can elapse without running system processes. If your IOS version does not support the **scheduler allocate** command, use this command instead.

```
r13(config)# scheduler interval 750
```

HANDLING CRASH DUMPS

If a router does crash, the core dump file should be sent to an FTP server (alternatively you can send this file via rcp or tftp depending on your IOS version) so the cause of the crash can possibly be determined. The router must be configured to send the file to an FTP server. It will need the IP address of the server, the username and password, and the file name to be created.

```
r13(config)# exception dump 192.168.1.100
r13(config)# exception protocol ?
  ftp    FTP protocol
  rcp    RCP protocol
  tftp   TFTP protocol

r13(config)# exception protocol ftp
r13(config)# exception core-file r13-pod1.ccbootcamp.com
r13(config)# ip ftp username cisco
r13(config)# ip ftp password ccie
```

BLACK HOLE ROUTES

Black hole routes are used to route known invalid networks to a null interface. To configure black hole routes, create static routes to null0. Check with IANA[7] to verify current reserved network blocks as they change periodically.

Make sure your null0 interface is configured for `no ip unreachables`. Also, do not configure black hole routes with TCP Intercept as unpredictable results may occur.

```
r13(config)# interface null0
r13(config-if)# no ip unreachables

r13(config)# ip route 1.0.0.0 255.0.0.0 null0
r13(config)# ip route 2.0.0.0 255.0.0.0 null0
r13(config)# ip route 5.0.0.0 255.0.0.0 null0
r13(config)# ip route 7.0.0.0 255.0.0.0 null0
r13(config)# ip route 10.0.0.0 255.0.0.0 null0
r13(config)# ip route 23.0.0.0 255.0.0.0 null0
r13(config)# ip route 27.0.0.0 255.0.0.0 null0
r13(config)# ip route 31.0.0.0 255.0.0.0 null0
r13(config)# ip route 36.0.0.0 255.0.0.0 null0
r13(config)# ip route 37.0.0.0 255.0.0.0 null0
r13(config)# ip route 39.0.0.0 255.0.0.0 null0
r13(config)# ip route 41.0.0.0 255.0.0.0 null0
r13(config)# ip route 42.0.0.0 255.0.0.0 null0
r13(config)# ip route 49.0.0.0 255.0.0.0 null0
r13(config)# ip route 50.0.0.0 255.0.0.0 null0
r13(config)# ip route 58.0.0.0 255.0.0.0 null0
r13(config)# ip route 59.0.0.0 255.0.0.0 null0
r13(config)# ip route 60.0.0.0 255.0.0.0 null0
r13(config)# ip route 69.0.0.0 255.0.0.0 null0
r13(config)# ip route 70.0.0.0 255.0.0.0 null0
r13(config)# ip route 71.0.0.0 255.0.0.0 null0
r13(config)# ip route 72.0.0.0 255.0.0.0 null0
r13(config)# ip route 73.0.0.0 255.0.0.0 null0
r13(config)# ip route 74.0.0.0 255.0.0.0 null0
r13(config)# ip route 75.0.0.0 255.0.0.0 null0
r13(config)# ip route 76.0.0.0 255.0.0.0 null0
r13(config)# ip route 77.0.0.0 255.0.0.0 null0
r13(config)# ip route 78.0.0.0 255.0.0.0 null0
r13(config)# ip route 79.0.0.0 255.0.0.0 null0
r13(config)# ip route 82.0.0.0 255.0.0.0 null0
r13(config)# ip route 83.0.0.0 255.0.0.0 null0
```

[7] http://www.iana.org/assignments/ipv4-address-space

```
r13(config)# ip route 84.0.0.0 255.0.0.0 null0
r13(config)# ip route 85.0.0.0 255.0.0.0 null0
r13(config)# ip route 86.0.0.0 255.0.0.0 null0
r13(config)# ip route 87.0.0.0 255.0.0.0 null0
r13(config)# ip route 88.0.0.0 255.0.0.0 null0
r13(config)# ip route 89.0.0.0 255.0.0.0 null0
r13(config)# ip route 90.0.0.0 255.0.0.0 null0
r13(config)# ip route 91.0.0.0 255.0.0.0 null0
r13(config)# ip route 92.0.0.0 255.0.0.0 null0
r13(config)# ip route 93.0.0.0 255.0.0.0 null0
r13(config)# ip route 94.0.0.0 255.0.0.0 null0
r13(config)# ip route 95.0.0.0 255.0.0.0 null0
r13(config)# ip route 96.0.0.0 255.0.0.0 null0
r13(config)# ip route 97.0.0.0 255.0.0.0 null0
r13(config)# ip route 98.0.0.0 255.0.0.0 null0
r13(config)# ip route 99.0.0.0 255.0.0.0 null0
r13(config)# ip route 100.0.0.0 255.0.0.0 null0
r13(config)# ip route 101.0.0.0 255.0.0.0 null0
r13(config)# ip route 102.0.0.0 255.0.0.0 null0
r13(config)# ip route 103.0.0.0 255.0.0.0 null0
r13(config)# ip route 104.0.0.0 255.0.0.0 null0
r13(config)# ip route 105.0.0.0 255.0.0.0 null0
r13(config)# ip route 106.0.0.0 255.0.0.0 null0
r13(config)# ip route 107.0.0.0 255.0.0.0 null0
r13(config)# ip route 108.0.0.0 255.0.0.0 null0
r13(config)# ip route 109.0.0.0 255.0.0.0 null0
r13(config)# ip route 110.0.0.0 255.0.0.0 null0
r13(config)# ip route 111.0.0.0 255.0.0.0 null0
r13(config)# ip route 112.0.0.0 255.0.0.0 null0
r13(config)# ip route 113.0.0.0 255.0.0.0 null0
r13(config)# ip route 114.0.0.0 255.0.0.0 null0
r13(config)# ip route 115.0.0.0 255.0.0.0 null0
r13(config)# ip route 116.0.0.0 255.0.0.0 null0
r13(config)# ip route 117.0.0.0 255.0.0.0 null0
r13(config)# ip route 118.0.0.0 255.0.0.0 null0
r13(config)# ip route 119.0.0.0 255.0.0.0 null0
r13(config)# ip route 120.0.0.0 255.0.0.0 null0
r13(config)# ip route 121.0.0.0 255.0.0.0 null0
r13(config)# ip route 122.0.0.0 255.0.0.0 null0
r13(config)# ip route 123.0.0.0 255.0.0.0 null0
r13(config)# ip route 124.0.0.0 255.0.0.0 null0
r13(config)# ip route 125.0.0.0 255.0.0.0 null0
r13(config)# ip route 126.0.0.0 255.0.0.0 null0
r13(config)# ip route 127.0.0.0 255.0.0.0 null0
r13(config)# ip route 169.254.0.0 255.255.0.0 null0
r13(config)# ip route 172.16.0.0 255.240.0.0 null0
r13(config)# ip route 192.0.0.0 255.255.128.0 null0
r13(config)# ip route 192.168.0.0 255.255.0.0 null0
r13(config)# ip route 197.0.0.0 255.0.0.0 null0
r13(config)# ip route 201.0.0.0 255.0.0.0 null0
r13(config)# ip route 221.0.0.0 255.0.0.0 null0
r13(config)# ip route 222.0.0.0 255.0.0.0 null0
r13(config)# ip route 223.0.0.0 255.0.0.0 null0
```

TYPICAL GOTCHAS!

- Blocking ICMP which prevents reachability testing
- Forgetting which attacks use ICMP echo, fragments, directed broadcasts, spoofed addresses, etc.
- Remembering to specify an access-list for RPF, if no access-list is applied all traffic that does not have a verifiable reverse path is dropped
- Forgetting that the access-list for TCP Intercept must specify TCP, not IP, for the protocol

CHAPTER 14

ACCESS LISTS

Access-lists are used for a variety of purposes. They are used to filter user traffic, control routing updates, and other miscellaneous security implications such as disabling IDS signatures, VPN phase II, controlling NAT, router management through interfaces (VTY, console, aux, etc.), and SNMP management.

General Rules of Access-Lists

- Can be named or numbered depending on the protocol
- Numbered ACL's must fall in the range assigned to the protocol
- Implicit deny at the end
- Once a packet matches an ACL, the router does not continue to process the rest of the lines (packet is either permitted or dropped depending on the match)
- Additional lines added to an ACL are put at the end of the list
- ACL's have no effect until they are applied
- Only one ACL per-interface per-direction
- Up to two ACL's total on an interface
- IOS ACL's use wildcard masks
- PIX ACL's use network masks

Recommendations

- Use Notepad when creating access-lists (delete and re-add the ACL after you make changes)
- Use descriptive names instead of numbers whenever possible
- Use remarks to further explain your ACL's
- Log denied packets (use the log-input option when available)
- Always use wildcard masks with standard ACL's
- Use ACL's with debugs to limit the information logged

Standard

Standard IP access-lists must be within the range of 1-99 or 1300-1999 if you are running IOS version 12.0.1 or later. You can only specific a network and a mask. You cannot specify any protocols or use a combination of source and destination addresses. Extended access-lists are required for either of those two options.

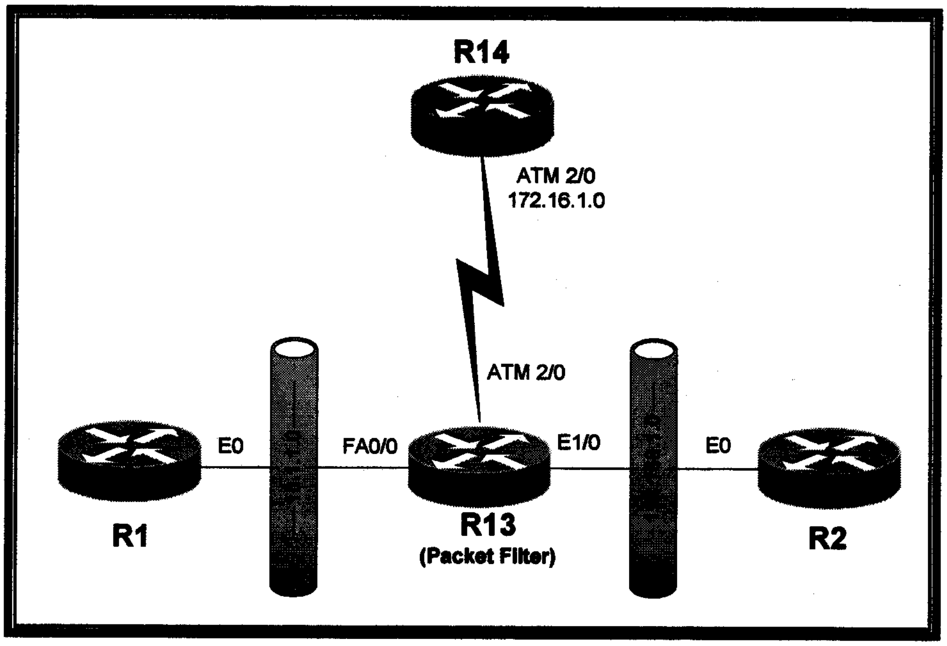
```
r13(config)# access-l ?
  <1-99>            IP standard access list
  <100-199>         IP extended access list
  <1000-1099>       IPX SAP access list
  <1100-1199>       Extended 48-bit MAC address access list
  <1200-1299>       IPX summary address access list
  <1300-1999>       IP standard access list (expanded range)
  <200-299>         Protocol type-code access list
  <2000-2699>       IP extended access list (expanded range)
```

```
<300-399>         DECnet access list
<400-499>         XNS standard access list
<500-599>         XNS extended access list
<600-699>         Appletalk access list
<700-799>         48-bit MAC address access list
<800-899>         IPX standard access list
<900-999>         IPX extended access list
dynamic-extended  Extend the dynamic ACL abolute timer
rate-limit        Simple rate-limit specific access list
```

STANDARD ACL'S ARE TYPICALLY USED FOR THE FOLLOWING:

- Basic packet filtering
- Routing protocol redistribution
- Controlling access to router ports (VTY, Console, Aux)
- Controlling SNMP access

Figure 14.1



In Figure 14.1 we have a router acting as a packet filter and RIP running between all 4 routers to give us connectivity. If we want to prevent one router from talking to the other we can block packets with a standard access-list and apply it to the interface. A standard access-list can be applied to either inbound or outbound. Standard ACL's when applied using access-groups are for source network only. An access-group applied inbound will permit or deny the source traffic. An access-group applied outbound will permit or deny the source from reaching that destination. In other words, inbound access-groups block traffic from even entering the router. Outbound access-groups block traffic from leaving the router's interface. So, if you look at Figure x.x we can prevent the 10.1.1.0 network from communicating with R13 or any hosts/routers behind it using an inbound access-group. We can also prevent network 10.1.1.0 from sending traffic to R2 without affecting any other interfaces using outbound access-groups.

INBOUND BLOCKING

R13 should be configured to block all traffic from network 10.1.1.0. Be aware that certain routing protocols will still function. In our example, routing is maintained throughout the small network because R13 has connected routes to all 3 routers. If you add another loopback on to R1 that route will be blocked. Remember that it is blocking inbound traffic only. Therefore, R13 will still be able to send routing updates to the other 3 routers.

Whenever we test access-lists using pings we always run `debug ip icmp` on all routers. This helps us to see if traffic makes it through one-way but the reply is not received for example[8].

```
r13(config)# access-list 1 deny 10.1.1.0 0.0.0.255 log
r13(config)# access-list 1 permit any
r13(config)# interface fa0/0
r13(config-if)# ip access-group 1 in

r1# ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
U
1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13.U
1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13.U
Success rate is 0 percent (0/5)
r1#
1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13
r1#
```

When we try and ping R2 we receive administratively prohibited unreachable messages from R13. These can be disabled on R13 using the command below. R1 no longer receives any indication of why the packet was dropped. This can be useful for IOS firewalls.

```
r13(config)# interface fa0/0
r13(config-if)# no ip unreachables

r1# ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
r1#
```

OUTBOUND BLOCKING

R13 should be configured to block all traffic from network 10.1.1.0 to the ATM network only. Traffic to the 192.168.1.0 network will be unaffected. The general rule is to block as close to the source address as possible, but using only standard access-lists we have to allow the traffic to traverse the FA0/0 interface before it gets blocked by the ATM interface. Typically, an extended ACL at the FA0/0 interface is a better option.

```
r1# ping 172.16.1.14

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.14, timeout is 2 seconds:
!!!!!
```

[8] This is also useful when debugging VPN's as it is fairly common to have traffic be encrypted one direction but no reply.

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/8/12 ms
r1#
1w1d: ICMP: echo reply rcvd, src 172.16.1.14, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 172.16.1.14, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 172.16.1.14, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 172.16.1.14, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 172.16.1.14, dst 10.1.1.1

r13(config)# interface atm2/0.1
r13(config-if)# ip access-group 1 out

r1# ping 172.16.1.14

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.14, timeout is 2 seconds:
U
1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13.U

1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13.U
Success rate is 0 percent (0/5)
r1#
1w1d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13

r1# ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/12/12 ms
r1#
1w1d: ICMP: echo reply rcvd, src 192.168.1.2, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 192.168.1.2, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 192.168.1.2, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 192.168.1.2, dst 10.1.1.1
1w1d: ICMP: echo reply rcvd, src 192.168.1.2, dst 10.1.1.1
```

EXTENDED

Extended ACL's must be within the range of 100-199 or 2000-2699 if you are running IOS version 12.0.1 or later. Alternatively, you can use named access-lists to give a descriptive name.

Extended ACL's are typically used for the following:

- Advanced packet filtering requiring both source and destination and/or specific protocols
- Routing protocol redistribution
- Controlling NAT
- Determining which packets will be encrypted for a VPN
- Defining interesting traffic for ISDN

In the previous example for standard access-lists we blocked traffic from the 10.1.1.0 network going to the 172.16.1.0 network. We did this at the ATM interface which would waste router resources. The more granular approach is usually a better option.

There are several steps when creating extended access-lists. First you must determine the traffic you will want to permit and/or deny. Then you can begin the actual process of creating the ACL.

Step 1. **Configure the access-list number.**

```
r13(config)# access-list 100 ?
  deny      Specify packets to reject
  dynamic   Specify a DYNAMIC list of PERMITs or DENYs
  permit    Specify packets to forward
  remark    Access list entry comment
```

Step 2. **Configure the protocol for the access-list line.**

```
r13(config)# access-list 100 deny ?
  <0-255>   An IP protocol number
  ahp       Authentication Header Protocol
  eigrp     Cisco's EIGRP routing protocol
  esp       Encapsulation Security Payload
  gre       Cisco's GRE tunneling
  icmp      Internet Control Message Protocol
  igmp      Internet Gateway Message Protocol
  igrp      Cisco's IGRP routing protocol
  ip        Any Internet Protocol
  ipinip    IP in IP tunneling
  nos       KA9Q NOS compatible IP over IP tunneling
  ospf      OSPF routing protocol
  pcp       Payload Compression Protocol
  pim       Protocol Independent Multicast
  tcp       Transmission Control Protocol
  udp       User Datagram Protocol
```

Step 3. **Configure the source address, host, or any.**

```
r13(config)# access-list 100 deny tcp ?
  A.B.C.D   Source address
  any       Any source host
  host      A single source host
```

Step 4. **Configure the destination address, host, or any.**

```
r13(config)# access-list 100 deny tcp any ?
  A.B.C.D   Destination address
  any       Any destination host
  eq        Match only packets on a given port number
  gt        Match only packets with a greater port number
  host      A single destination host
  lt        Match only packets with a lower port number
  neq       Match only packets not on a given port number
  range     Match only packets in the range of port numbers
```

Step 5. **(Optional) Configure any additional options. Below we highlighted the most common options that you should be familiar with.**

```
r13(config)# access-list 100 deny tcp any any ?
  ack          Match on the ACK bit
  dscp         Match packets with given dscp value
  eq           Match only packets on a given port number
  established  Match established connections
  fin          Match on the FIN bit
  fragments    Check non-initial fragments
  gt           Match only packets with a greater port number
  log          Log matches against this entry
  log-input    Log matches against this entry, including input interface
  lt           Match only packets with a lower port number
  neq          Match only packets not on a given port number
  precedence   Match packets with given precedence value
  psh          Match on the PSH bit
  range        Match only packets in the range of port numbers
  rst          Match on the RST bit
  syn          Match on the SYN bit
  time-range   Specify a time-range
  tos           Match packets with given TOS value
  urg           Match on the URG bit
  <cr>
```

Step 6. (Optional) Select any sub-options. With this example, we are choosing to deny telnet packets from any source to any destination. If you select a port number that the router has a name for, it will substitute the name in the configuration file.

```
r13(config)# access-list 100 deny tcp any any eq ?
  <0-65535>     Port number
  bgp           Border Gateway Protocol (179)
  chargen       Character generator (19)
  cmd           Remote commands (rcmd, 514)
  daytime       Daytime (13)
  discard       Discard (9)
  domain        Domain Name Service (53)
  echo          Echo (7)
  exec          Exec (rsh, 512)
  finger        Finger (79)
  ftp           File Transfer Protocol (21)
  ftp-data      FTP data connections (used infrequently, 20)
  gopher        Gopher (70)
  hostname      NIC hostname server (101)
  ident         Ident Protocol (113)
  irc           Internet Relay Chat (194)
  klogin        Kerberos login (543)
  kshell        Kerberos shell (544)
  login         Login (rlogin, 513)
  lpd           Printer service (515)
  nntp          Network News Transport Protocol (119)
  pim-auto-rp   PIM Auto-RP (496)
  pop2          Post Office Protocol v2 (109)
  pop3          Post Office Protocol v3 (110)
  smtp          Simple Mail Transport Protocol (25)
  sunrpc        Sun Remote Procedure Call (111)
  syslog        Syslog (514)
  tacacs        TAC Access Control System (49)
  talk          Talk (517)
  telnet        Telnet (23)
  time          Time (37)
  uucp          Unix-to-Unix Copy Program (540)
  whois         Nicname (43)
  www           World Wide Web (HTTP, 80)

r13(config)# access-list 100 deny tcp any any eq 23

r13# show access-list 100
Extended IP access list 100
    deny tcp any any eq telnet
```

Commented Entries

Commented entries are useful when you have access-lists that you need to quickly understand what they are for. Long ACLs or a router with several ACLs are the best situation to use comments.

```
r13(config)# access-list 100 remark Deny Telnet traffic
r13(config)# access-list 100 deny   tcp any any eq telnet
r13(config)# access-list 100 remark Permit all traffic
r13(config)# access-list 100 permit ip any any
```

Here is how it will look in the configuration. Notice that the lines are created exactly as entered. Also, you are not limited to one remark per ACL. You can have remarks throughout the list.

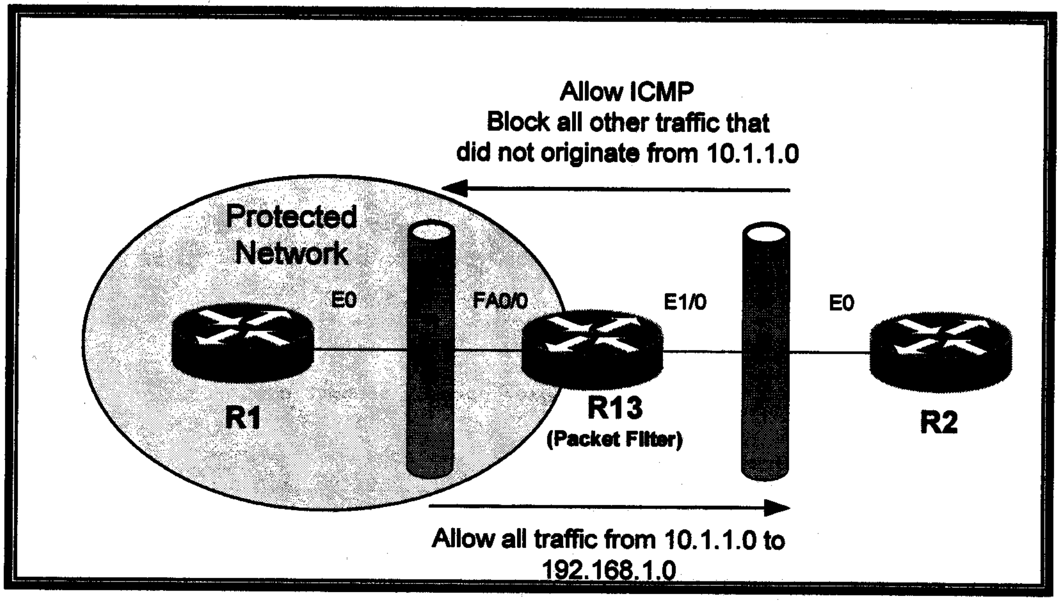
```
access-list 100 remark Deny Telnet traffic
access-list 100 deny tcp any any eq telnet
```

```
access-list 100 remark Permit all traffic
access-list 100 permit ip any any
```

REFLEXIVE

Reflexive ACLs allow for dynamic ACLs that are changed based on user traffic. Essentially it operates They were primarily used prior to the availability of the IOS Firewall feature set. However, if you are given the following question for example, you would have to know how to configure reflexive ACLs.

Figure 14.2 *Reflexive ACL example*


Allow ICMP
Block all other traffic that
did not originate from 10.1.1.0
Protected
Network
E0
FA0/0
E1/0
E0
R1
R13
(Packet Filter)
R2
Allow all traffic from 10.1.1.0 to
192.168.1.0

Question: Configure R13 as a firewall. Allow TCP traffic to be initiated from R1 to R2, but not vice versa. The connection must be closed after 60 seconds of inactivity and the ACL entry removed. You may not use the `ip inspect command`. Allow ICMP pings to pass through R13.

Step 1. Create ACLs for inbound traffic. Remember that inbound traffic actually refers to outbound traffic from the perspective of the hosts on the sending network. These hosts would be on what we would call the "protected network." It is protected because dynamic ACLs are created for traffic sent by hosts on this network.

```
r13(config)# ip access-list extended OUTBOUND
r13(config-ext-nacl)# permit ip any any reflect tcptraffic
```

Step 2. Create ACLs for outbound traffic. This is traffic coming from unprotected networks. It will "evaluate" incoming traffic to see if the ACL will allow it to pass. Dynamic entries are created as packets pass through the ACL that is set to "reflect." In other words, reflect creates the dynamic entries and "evaluate" checks incoming packets to make sure there is in fact an ACL that allows it to pass.

```
r13(config)# ip access-list extended INBOUND
```

```
r13(config-ext-nacl)# permit icmp 192.168.1.0 0.0.0.255 10.1.1.0 0.0.0.255
echo
r13(config-ext-nacl)# evaluate tcptraffic
```

Step 3. Apply the ACL's to the interface.

```
r13(config)# interface fa0/0
r13(config-if)# ip access-group OUTBOUND in
r13(config-if)# ip access-group INBOUND out
```

Step 4. (Optional) Configure the timeout. The default is 300 seconds.

```
r13(config)# ip reflexive-list timeout 60
```

Step 5. Verify your configuration by attempting to send traffic in each direction. Make sure that packets you expect to be permitted get through and those that should not be permitted get denied. You should also see dynamic ACL entries under Reflexive Access Lists.

```
r1# telnet 192.168.1.2
Trying 192.168.1.2 ... Open


User Access Verification

Password: cisco
r2>

r2# telnet 10.1.1.1
Trying 10.1.1.1 ...
% Destination unreachable; gateway or host down

r13# show access-list
Extended IP access list INBOUND
    permit icmp 192.168.1.0 0.0.0.255 10.1.1.0 0.0.0.255 echo
    evaluate tcptraffic
Extended IP access list OUTBOUND
    permit ip any any reflect tcptraffic
Reflexive IP access list tcptraffic
    permit tcp host 192.168.1.2 eq telnet host 10.1.1.1 eq 11009 (21
matches) (time left 57)
    permit udp host 255.255.255.255 eq rip host 10.1.1.1 eq rip (221
matches) (time left 58)
```

TIME-BASED

In the previous example, we allowed hosts on network 10.1.1.0 to telnet to hosts on 192.168.1.0. In this example we will only allow them to telnet during regular business hours (9am – 5pm) Monday through Friday.

Step 1. Configure the time range name.

```
r13(config)# time-range REGULAR_BUSINESS_HOURS
r13(config-time-range)#?
Time range configuration commands:
  absolute   absolute time and date
  default    Set a command to its defaults
  exit       Exit from time-range configuration mode
  no         Negate a command or set its defaults
  periodic   periodic time and date
```

Step 2. Configure the type of time range. Absolute would be useful for a contractor, for example, that will have his contract expire in say 30 days. This would allow you to create access that will expire along with their contract. It is similar to what you would see when system administrators create accounts for temporary workers or contractors. When using a time of day, remember to use military time. Also, it is recommended to use NTP if possible. For the exam, set your system clock to the local time.

```
r13(config-time-range)# periodic ?
  Friday     Friday
  Monday     Monday
  Saturday   Saturday
  Sunday     Sunday
  Thursday   Thursday
  Tuesday    Tuesday
  Wednesday  Wednesday
  daily      Every day of the week
  weekdays   Monday thru Friday
  weekend    Saturday and Sunday

r13(config-time-range)# periodic weekdays ?
  hh:mm  Starting time

r13(config-time-range)# periodic weekdays 09:00 to 17:00
```

Step 3. Add the time-range option to an ACL.

```
r13(config)# access-list 100 permit tcp 10.1.1.0 0.0.0.255 192.168.1.0
0.0.0.255 eq 23 time-range REGULAR_BUSINESS_HOURS
```

Step 4. Set the system clock or configure NTP.

```
r13# clock set 16:15:00 4 June 2003
```

Step 5. Apply the ACL to an interface.

```
r13(config)# interface fa0/0
r13(config-if)# ip access-group 100 in
```

Step 6. Verify your configuration. The first test occurred at 4:50pm. The next test was at 5:05pm. Despite what the lab directions may say, you should always change your clock so that you can verify your configuration. Simply change it back to the proper time after you finish your test.

```
r1# telnet 192.168.1.2
Trying 192.168.1.2 ... Open


User Access Verification

Password: cisco
r2>

r1# telnet 192.168.1.2
Trying 192.168.1.2 ...
% Destination unreachable; gateway or host down

r1#
1w3d: ICMP: dst (10.1.1.1) administratively prohibited unreachable rcv
from 10.1.1.13
r1#
```

DYNAMIC (LOCK AND KEY)

Lock and Key allows an administrator to require a user to authenticate to the router before the user's traffic is permitted. The user must first telnet to the router and authenticate to either the local database or a AAA server. The router will then create a dynamic ACL to allow the traffic. This is basically just like reflexive ACLs with the additional security of requiring authentication.

The configuration that works with 12.1.5T9 is not the same as may work for other versions. The command set on 12.1.5T9 appears to be missing some commands.

CONFIGURATION FOR 12.1.5T9

Step 1. Configure the AAA method you want to use. It is recommended to always start with local authentication prior to configuring the router to use an external security server. It is much easier to troubleshoot when everything occurs locally on the router.

```
r13(config)# username cisco password ccie
```

Step 2. Configure the VTY for the access-enable autocommand. This will force the router to disconnect you after you Telnet to the router and open the dynamic ACL[9].

```
r13(config)# username cisco autocommand access-enable host timeout 5
```

Step 3. Configure at least 2 ACLs. The first ACL is to allow the users to Telnet to the router. The second and remaining ACLs are to allow the desired traffic through the router once the user is authenticated.

```
r13(config)# access-list 101 permit tcp any host 10.1.1.13 eq telnet
r13(config)# access-list 101 dynamic CCIE timeout 2 permit ip 10.1.1.0
0.0.0.255 192.168.1.0 0.0.0.255
```

Step 4. Verify your configuration by Telnetting to the firewall router and checking that it automatically disconnects your session. Then you should be able to Telnet or send other traffic through the firewall.

```
r1# telnet 10.1.1.13
Trying 10.1.1.13 ... Open


User Access Verification

Username: cisco
Password: ccie

[Connection to 10.1.1.13 closed by foreign host]

r1# telnet 192.168.1.2
Trying 192.168.1.2 ... Open


User Access Verification

Password:
r2>
```

CONFIGURATION FOR OTHER VERSIONS

Step 1. Configure the AAA method you want to use. It is recommended to always start with local authentication prior to configuring the router to use an external security server. It is much easier to troubleshoot when everything occurs locally on the router.

```
r13(config)# username cisco password ccie
```

Step 2. Configure the router to disconnect the telnet session once the user authenticates.

```
r13(config)# username cisco autocommand access-enable host timeout 5
```

Step 3. Configure the VTY for local login.

```
r13(config-line)# line vty 0 4
r13(config-line)# login local
```

Step 4. Configure at least 2 ACLs. The first ACL is to allow the users to Telnet to the router. The second and remaining ACLs are to allow the desired traffic through the router once the user is authenticated.

```
r13(config)# access-list 101 permit tcp any host 10.1.1.13 eq telnet
r13(config)# access-list 101 dynamic CCIE timeout 2 permit ip 10.1.1.0
0.0.0.255 192.168.1.0 0.0.0.255
```

ACL'S AND DEBUGS

When running debug the router may become overwhelmed and stop responding. During the lab, this is not likely to be a problem. However, too much information makes it difficult to read all the lines of debug. Considering the time sensitive nature of the exam, you should be proficient in controlling your debugs to give you the exact output you want.

Step 1. Disable fast switching on any interfaces that will be involved in gathering debug information. If fast switching is enabled, you will only see the first packet. Be aware of this behavior in case this is desired.

```
r13(config)# interface fa0/0
r13(config-if)# no ip route-cache
```

Step 2. Create a specific ACL that will limit the traffic to only what you want to see.

```
r13(config)# access-list 110 permit udp any any eq rip
```

Step 3. Run your debug and specify the ACL. Note that some debugs particularly with BGP allow you to set an ACL. Otherwise just use `debug ip packet <acl_number>` or `debug ip packet <acl_number>` detail.

```
r13# debug ip packet detail 110
IP packet debugging is on (detailed) for access list 110
```

Step 4. Disable the debug by typing `no debug 'x'` where 'x' is exactly what you typed to start the debug process. Alternatively, type `no debug all` or `u all`.

```
r13# u all
All possible debugging has been turned off
```

TYPICAL GOTCHAS!

- Forgetting the implicit deny at the end of an ACL
- Not logging ACL deny traffic
- Forgetting to reapply an access-group to an interface if the ACL is deleted
- Using network masks instead of wildcard masks
- Adding lines to existing ACL's unintentionally
- Not using ACL's with debugs and overloading the router

CHAPTER 15

IOS FIREWALL

The IOS Firewall feature set is available as an optional, add-on software version of IOS that has additional functionality similar to a PIX firewall as well as intrusion detection. It is mainly used when a traditional firewall cannot be used because of budget or other restraints. Most Cisco routers equipped with enough memory and processing power have the ability to run the IOS Firewall Feature set.

The IOS Firewall has three main features. These features are used with Context-Based Access Control (CBAC). Additionally, Intrusion Detection is an add-on module for the Firewall Feature set.

- Traffic Filtering
- Traffic Inspection
- Alerts and Audit Trails

TRAFFIC FILTERING

CBAC filters TCP and UDP packets at the application layer. Using an intelligent, stateful filtering process, CBAC filters traffic flowing through the router. The filtering uses a combination of manually configured access-lists as well as dynamic access-lists created by the traffic inspection process. CBAC uses both traffic filtering and traffic inspection to allow packets in and out of the router.

TRAFFIC INSPECTION

CBAC only inspects the protocols that you specify. It inspects at the network, transport, and even application layer for some traffic types. Packets entering the firewall are inspected by CBAC only if they first pass the inbound access list at the interface. Packets that do not travel through the firewall are not inspected. It does not inspect traffic that it observes on the LAN.

An IOS firewall does not have a set "inside" or "outside" interface. CBAC knows which packets to protect by configuring an interface to be protected and analyzing which direction the traffic is flowing. Typically, CBAC is configured to allow traffic out of the protected interface and only permit valid return traffic. Access lists are dynamically changed to allow the return traffic back in based on the state information gathered in the inspection process.

CBAC maintains a state table for TCP connections to determine if incoming traffic from the outside is a valid return packet. UDP traffic is approximated based on the idle time configured. CBAC also has the ability to handle multiple channels and dynamic ports that are dynamically created when using multimedia applications and other protocols such as FTP, RPC, and SQLNet.

It also inspects and monitors only the control channels of connections for applications such as FTP.

CBAC also has the ability to detect and prevent various types of DoS attacks. Suspicious packets or packets with TCP sequence numbers outside of the expected range are dropped. TCP SYN floods can also be limited. CBAC can be configured to allow only a certain number of half-open connections to exist as well as limit the rate of half-opens per minute. Do not confuse this capability with TCP Intercept or TCP Watch. Although the results are similar, the commands are not the same.

Fragmented IP packets used in DoS attacks are also prevented by CBAC. A host that receives fragmented packets will have to use resources to attempt to reassemble the incomplete packets. Fragmented IP packets are illegally fragmented or intentionally incomplete. CBAC inspection recognizes these packets and does not permit them to be sent to the intended host.

Alerts and Audit Trails

CBAC generates real-time alerts when there is an attempt to violate the security rules. Alerts are sent to the console, buffer, or Syslog server depending on how your logging is configured. Alerts can be configured globally for all protocols or for specific protocols only. The following is a sample alert log message from a potential DoS attack.

```
%FW-4-ALERT_ON: getting aggressive, count (550/500) current 1-min rate: 250
%FW-4-ALERT_OFF: calming down, count (0/400) current 1-min rate: 0
```

Audit trails track details of communications such as source IP address, source ports, destination IP address, destination port, bytes sent, and recording timestamps. Audit trails use the same logging features as alerts. Audit-trails can be configured globally for all protocols or for specific protocols only. The following is a sample log from an audit-trail.

```
%FW-6-SESS_AUDIT_TRAIL: tcp session initiator (192.168.1.13:33192) sent 22
    bytes -- responder (192.168.129.11:25) sent 208 bytes
%FW-6-SESS_AUDIT_TRAIL: http session initiator (172.16.57.30:44673) sent 1599
    bytes -- responder (172.21.127.218:80) sent 93124 bytes
```

Configuring CBAC

Configuring a Basic Two-Port Firewall

Step 1 Create an inspection rule(s) that specifies which protocols are protected by CBAC. Our rule is called "FW." The name of the rule is arbitrary.

```
r13(config)# ip inspect name FW ?
cuseeme        CUSeeMe Protocol
fragment       IP fragment inspection
ftp            File Transfer Protocol
h323           H.323 Protocol (e.g., MS NetMeeting, Intel Video Phone)
http           HTTP Protocol
netshow        Microsoft NetShow Protocol
rcmd           R commands (r-exec, r-login, r-sh)
realaudio      Real Audio Protocol
rpc            Remote Procedure Call Protocol
rtsp           Real Time Streaming Protocol
smtp           Simple Mail Transfer Protocol
sqlnet         SQL Net Protocol
streamworks    StreamWorks Protocol
tcp            Transmission Control Protocol
```

```
tftp            TFTP Protocol
udp             User Datagram Protocol
vdolive         VDOLive Protocol
```

After selecting the protocol(s) you want to inspect, there are several options to configure. Alerts will send log notifications. Alerts are enabled for all configured protocols by default. Audit-trails allow the router to log details about a session. It can be enabled for individual protocols or for all CBAC configured protocols by entering the **ip inspect audit-trail** command. By default, audit-trail is disabled. The timeout for TCP is 3600 seconds (1 hour). If a timeout value is not set, the default value for that protocol will be used. Below are a few examples of frequently inspected protocols.

```
r13(config)# ip inspect name FW tcp ?
  alert          Turn on/off alert
  audit-trail    Turn on/off audit trail
  timeout        Specify the inactivity timeout time
  <cr>

r13(config)# ip inspect name FW tcp alert on audit-trail on timeout 600
r13(config)# ip inspect name FW udp alert on audit-trail on timeout 30
r13(config)# ip inspect name FW fragment maximum 500 timeout 30
```

Step 2 Configure an inbound access-list and apply it to the protected interface. This access list is often very general and usually is configured to just allow the local network(s) to communicate with other networks. In our example, we are allowing TCP, UDP, and ICMP protocols with a source address from 192.168.1.0 /24 subnet.

```
r13(config)# access-list 100 permit tcp 192.168.1.0 0.0.0.255 any
r13(config)# access-list 100 permit udp 192.168.1.0 0.0.0.255 any
r13(config)# access-list 100 permit icmp 192.168.1.0 0.0.0.255 any
r13(config)# access-list 100 permit deny ip any any log-input
```

Step 3 Apply the access-list to the interface. All traffic coming in to the router's ethernet interface from the LAN will have to pass access list 100.

```
r13(config)# interface ethernet0
r13(config-if)# ip access-group 100 in
```

Step 4 Configure an inbound access-list and apply it to the unprotected interface.

```
r13(config)# access-list 101 deny ip any any log
```

Step 5 Apply the access-list to the unprotected interface. This access-list can be as simple as a **deny ip any any**. This access list will be dynamically modified by CBAC. You must use extended access-lists.

```
r13(config)# interface serial0
r13(config-if)# ip access-group 101 in
```

Step 6 Apply the inspect rule to the protected interface. As traffic enters the interface, CBAC tracks the session information to determine how to dynamically open the access lists to permit the return traffic. In this case we want to have traffic coming from the Ethernet 0 LAN through the router be inspected. Since the router only has two interfaces, the configuration is simple. CBAC inspects packets inbound through the Ethernet 0 interface and inbound on the Serial 0 interface since that is where we applied access lists. The Ethernet 0 and Serial 0 interfaces do not have an outbound access-list so all traffic outbound is permitted. However, the Serial 0 does have an access-list with a **deny ip any any** configured. The access list on the Serial 0 interface is dynamically modified by

CBAC to permit the return traffic so even though there is a **deny ip any any**, CBAC will add additional lines to the access list.

```
r13(config)# interface ethernet0
r13(config-if)# ip inspect FW in
```

CBAC VERIFICATION

To see what CBAC is allowing back in, enter the **show access-list 101** command. Notice that it added additional permit lines above the deny any that we configured.

```
r13# show access-list 101
Extended IP access list 101
    permit tcp host 209.132.1.30 eq pop3 host 192.168.1.10 eq 3570
    permit tcp host 64.70.193.182 eq www host 192.168.1.10 eq 3595 (4 matches)
    permit tcp host 64.70.193.182 eq www host 192.168.1.10 eq 3594 (17
    matches)
    permit tcp host 64.70.193.182 eq www host 192.168.1.10 eq 3593 (8 matches)
    permit tcp host 64.94.89.210 eq www host 192.168.1.10 eq 3566 (8 matches)
permit udp host 207.217.126.41 eq domain host 192.168.1.200 eq 32772 (2
matches)
    deny ip any any (8 matches)
```

CBAC SESSION TIMERS AND THRESHOLD COMMANDS

Now that our router is configured for basic CBAC, we can configure optional session timers, threshold commands, fragment inspection, and Java applet filtering.

DOS PREVENTION RELATED COMMANDS

There are several commands to help limit TCP SYN attacks. The only IOS Firewall configurable DoS prevention commands are for TCP SYN attacks.

The command **ip inspect max-incomplete high <number>** defines the number of existing half-open sessions that will cause the IOS firewall to start deleting TCP half-open sessions. The default is 500.

```
r13(config)# ip inspect max-incomplete high 750
```

The command **ip inspect max-incomplete low <number>** defines the number of existing half-open sessions that will cause the IOS firewall to stop deleting half-open sessions. The default is 400.

```
r13(config)# ip inspect max-incomplete low 250
```

The command **ip inspect one-minute high <number>** defines the number of new TCP half-open sessions per minute. Once this number is reached, the oldest half-open sessions are deleted to make room for new connections. The default is 500 per minute.

```
r13(config)# ip inspect one-minute high 600
```

The command **ip inspect one-minute low <number>** defines the number of new TCP half-open sessions per minute that will cause the IOS firewall to stop deleting half-open connections. Once the IOS firewall exceeds the **ip inspect one-minute high** threshold, the oldest half-opens are deleted until the half-opens per minute is below the one-minute low threshold. The default is 400 per minute.

```
r13(config)# ip inspect one-minute low 300
```

The command **ip inspect tcp synwait-time <seconds>** defines the number of seconds the IOS firewall will wait for a half-open TCP session to be established before dropping the session. The default is 30 seconds.

```
r13(config)# ip inspect tcp synwait-time 15
```

Session Timeout Related Commands

By default, the IOS firewall will wait a fairly long time before it will terminate TCP and UDP connections. Frequently, the default values are too long for a secure environment. It may increase security to terminate idle connections at a faster rate. But, be aware that terminating these idle connections too soon may potentially cause problems with users having to reconnect. However, the defaults are still considered too long especially the TCP idle time.

The command **ip inspect tcp idle-time <seconds>** defines the number of seconds a TCP session may remain idle before the IOS Firewall terminates the session and remove it from the state table. The default is 3,600 seconds (1 hour).

```
r13(config)# ip inspect tcp idle-time 300
```

The command **ip inspect udp idle-time <seconds>** defines the number of seconds a UDP session may remain idle before the IOS Firewall terminates the session and remove it from the state table. The default is 30 seconds.

```
r13(config)# ip inspect udp idle-time 15
```

The command **ip inspect tcp finwait-time <seconds>** defines the number of seconds the IOS firewall will wait to terminate a TCP connection after the TCP-FIN exchange. The default is 5 seconds.

```
r13(config)# ip inspect tcp finwait-time 3
```

The command **ip inspect dns-timeout** specifies the DNS idle timeout. A host inside the IOS firewall sends DNS name lookup requests and a DNS server responds. The IOS firewall will only wait a certain amount of time for that response before it closes the temporary hole in the access-list. The default is 5 seconds.

```
r13(config)# ip inspect dns-timeout 3
```

Configuring IP Packet Fragmentation Inspection

CBAC can be configured to inspect IP fragments. These fragmented packets are frequently used in DoS attacks. If the first fragment is permitted to pass through the firewall, subsequent packets will also be permitted. However, if a fragment is received and is not an initial packet it will be dropped. This may cause a problem for legitimate traffic that happens to arrive out of order. Use this option with caution in a production environment.

The command **ip inspect name <name> fragment maximum <number> timeout <number>** defines the number of IP fragments the IOS firewall will permit and how long it will wait before they timeout. By default, fragments are not inspected.

```
r13(config)# ip inspect name FW fragment maximum 500 timeout 30
```

BLOCKING JAVA APPLETS

Java applets contain code that is downloaded and run on a client PC or host. These applets may contain malicious code that can be used by an attacker to compromise a system. The IOS firewall can block Java applets. However, Java applets wrapped in ZIP files or other archive type files cannot be blocked. Also, only Java applets downloaded via the HTTP protocol using TCP port 80 can be blocked. Java applets downloaded via FTP or HTTP using a non-standard port cannot be blocked. The following is a sample log message from a router that blocked a Java applet.

```
%FW-4-HTTP_JAVA_BLOCK: JAVA applet is blocked from (172.21.127.218:80) to
    (172.16.57.30:44673).
```

Java applets can also be blocked with CBAC. There are three options to deny Java applets.

- Block only specific sites known to contain malicious applets
- Block all sites except those specifically listed as permitted (trusted)
- Block all sites

It is recommended to block Java applets from all untrusted sources. To configure Java applet blocking, create a standard IP access list and an inspection rule that filters Java applets. In our example below, we allow only two specific sites for Java applets. The use of access lists permits us to be granular in how we permit or deny Java applets from specific sites based on their IP address.

```
router(config)# ip inspect name FW http java-list 10 timeout 3600
router(config)# access-list 10 permit 206.98.202.75
router(config)# access-list 10 permit 64.253.197.214
router(config)# access-list 10 deny any
```

PERMITTING TRAFFIC THROUGH THE IOS FIREWALL

We've discussed how to permit inside traffic to communicate with outside hosts, but what if we have web servers or mail servers on a DMZ type interface? Without NAT, we simply need to add access list permit statements for the access list applied to our Serial 0 interface.

NETWORK ADDRESS TRANLSATION (NAT)

NAT is used to map an inside local IP address to an outside local IP address. NAT is typically used for Internet connectivity when your internal network runs unregistered addresses as defined in RFC 1918.

Figure 15.1 *RFC 1918 - Unregistered IP addresses*

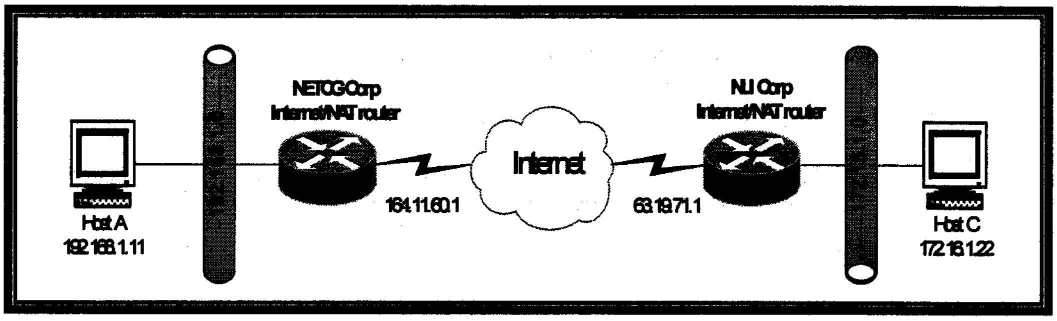
```
10.0.0.0 through 10.255.255.255
172.16.0.0 through 172.31.255.255
192.168.0.0 through 192.168.255.255
```

Cisco defines the different NAT IP address types as follows.

- Inside local address - The IP address assigned to a host on the inside network. The address is likely not a legitimate IP address assigned by the Network Information Center (NIC) or service provider.
- Inside global address - A legitimate IP address assigned by the NIC or service provider that represents one or more inside local IP addresses to the outside world.
- Outside local address - The IP address of an outside host as it appears to the inside network. Not necessarily a legitimate address, it is allocated from an address space routable on the inside.
- Outside global address - The IP address assigned to a host on the outside network by the host's owner. The address is allocated from a globally routable address or network space.

What is considered inside versus outside is simply a matter of what network belongs to us. In figure 16.2, let's assume we are the administrators for NETCG Corp. The hosts on our internal 192.168.1.0 network are inside local addresses. We map our internal hosts to an inside global address in the 164.11.60.0 network. Any addresses outside of our network are considered outside. Assuming NLI Corp is running NAT, their 63.19.71.0 network is the outside global address and the 172.16.1.0 network is the outside local address.

Figure 15.2 *NAT example*

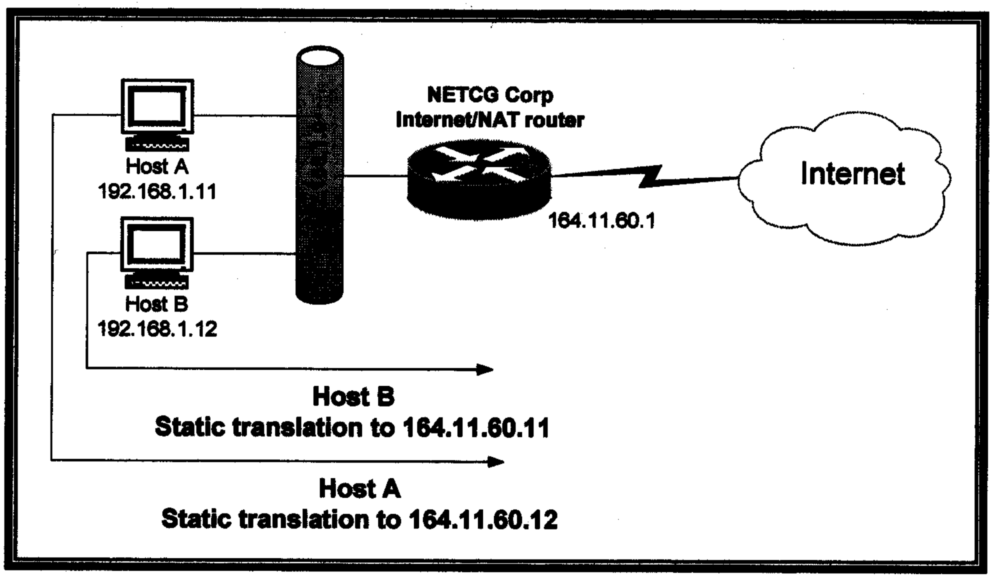


There are three types of address translation: static, dynamic, and port. Static translation establishes a one-to-one mapping between an inside local address and an inside global address. Static translations are useful when a host on the inside must be accessible by a fixed address from the outside. Web and mail servers that communicate directly with other Internet hosts usually must have a static NAT. Dynamic translation establishes a mapping between an inside local address and a pool of global addresses. Typically, an organizations PC's or workstations that need Internet access use dynamic translation since they do not need a fixed address. Port address translation establishes a mapping between many inside hosts and a single static address from the outside. This is the same as a dynamic translation except that instead of using a pool of addresses, there is only a single address. This single address can also be shared with the outside interface of the router.

Static Translation

In Figure 15.3, we have two hosts that we need to statically NAT. Host A is a web server and Host B is an SMTP mail server. If access-lists are applied to your interfaces, make sure you create additional access list lines to allow the necessary traffic.

Figure 15.3 *NAT – static translation*



Step 1 Configure NAT to map the inside local and outside local addresses for the hosts.

```
r13(config)# ip nat inside source static 192.168.1.11 164.11.60.11
r13(config)# ip nat inside source static 192.168.1.12 164.11.60.12
```

Step 2 Apply NAT to the inside interface.

```
r13(config)# interface ethernet 0/0
r13(config-if)# ip nat inside
```

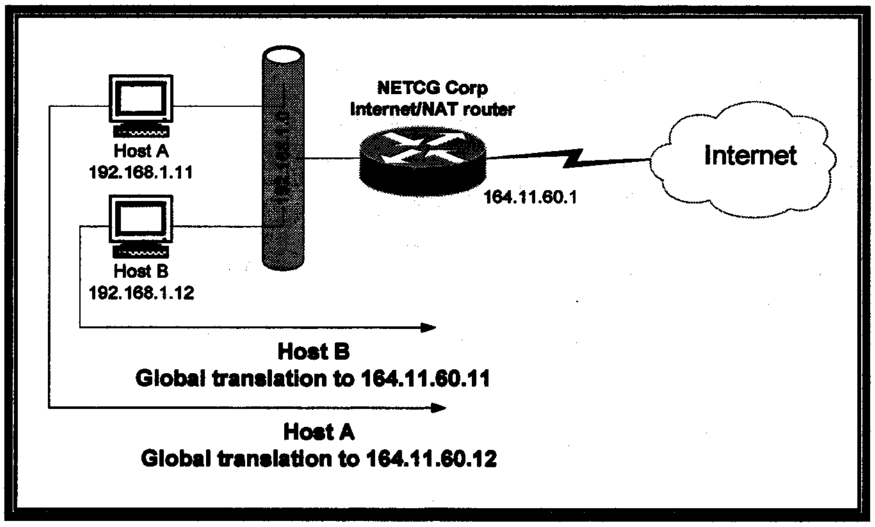
Step 3 Apply NAT to the outside interface.

```
r13(config)# interface serial 0/0
r13(config-if)# ip nat outside
```

Dynamic Translation

In Figure 15.4, we have several internal hosts that are mapped to a global pool of available addresses. Make sure that the addresses you use in your pool are available and unused.

Figure 15.4 *NAT – Dynamic translation*



Step 1 Create the pool of addresses to be used by NAT.

```
r13(config)# ip nat pool R13-POOL 164.11.60.10 164.11.60.254 netmask
255.255.255.0
```

Step 2 Create an access list that will be used to determine what addresses will be allowed for NAT.

```
r13(config)# access-list 1 permit 192.168.1.0 0.0.0.255
```

Step 3 Bind both the pool and the access list using the global `ip nat inside` command. This allows you to create multiple pools and assign different networks to different pools.

```
r13(config)# ip nat inside source list 1 pool R13-POOL
```

Step 4 Apply NAT to the inside interface.

```
r13(config)# interface ethernet 0/0
r13(config-if)# ip nat inside
```

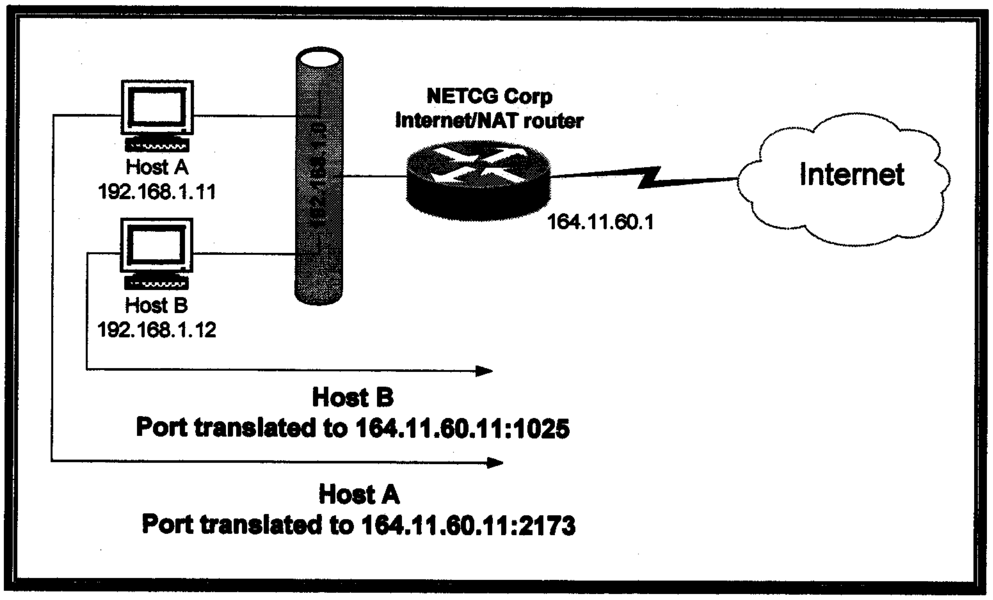
Step 5 Apply NAT to the inside interface.

```
r13(config)# interface serial 0/0
r13(config-if)# ip nat outside
```

PORT ADDRESS TRANSLATION (PAT)

PAT is used to map many inside addresses to a single global IP address. The router attaches session information to each connection in order to keep track of the communication. PAT typically does not work well with multimedia or voice connections.

Figure 15.5 *Port Address Translation*



In Figure 15.5, we have several internal hosts that are mapped to a single IP address. This is useful if you only have a single static IP address assigned by your provider. Typically, you need to use the same IP address as your interface. This is particularly useful if your provider assigns your IP address via DHCP

Step 1 Create an access list that will be used to determine what addresses will be allowed for NAT.

```
r13(config)# access-list 1 permit 192.168.1.0 0.0.0.255
```

Step 2 Configure NAT to use the access-list created in Step 1 and to use the interface serial 0/0 as the global address.

```
r13(config)# ip nat inside source list 1 interface serial 0/0
```

Step 3 Apply NAT to the inside interface.

```
r13(config)# interface ethernet 0/0
r13(config-if)# ip nat inside
```

Step 4 Apply NAT to the inside interface.

```
r13(config)# interface serial 0/0
r13(config-if)# ip nat outside
```

OUTSIDE NAT

Almost all implementations of NAT involve the inside address(es) being mapped to a global address(es). There may be a situation where we want to map the outside address to an inside address. This is accomplished using the **`ip nat outside`** command. The rest of the configuration remains the same. On an Internet router, this configuration is highly unlikely.

```
ip nat outside source static
```

- translates the source of the IP packets that are traveling outside to inside
- translates the destination of the IP packets that are traveling inside to outside

```
ip nat inside source static
```

- translates the source of IP packets that are traveling inside to outside
- translates the destination of the IP packets that are traveling outside to inside

IOS FIREWALL NOTES

Do not use CBAC with CEF since CBAC only works with process and fast switching. It does not inspect non-IP protocols including ICMP. Packets with the firewall's IP address as the source or destination are NOT inspected

TYPICAL GOTCHAS!

- Inspecting the wrong interfaces or direction
- Putting ACLs on the wrong interfaces or direction
- Missing required ACLs
- Expecting the "?" help to work
- Enabling audit-trails and generating too much log info

CHAPTER 16

PIX FIREWALL

PIX FEATURES

The PIX firewall uses stateful packet filtering which allows it to determine if a return packet is legitimate. It is similar to how the IOS Firewall dynamically opens and closes ports to allow return traffic that originated on the inside.

The PIX firewall uses the Adaptive Security Algorithm (ASA) which is the heart of the PIX operating system that allows for stateful inspection as well as other functions including randomizing TCP sequence numbers, port number, and other TCP flags. It also tracks UDP connections.

The PIX firewall also utilizes Cut-Through Proxy Operation which allows a user to authenticate and then have its traffic flow passed at the lower layers of the OSI model thereby improving performance

Lastly, the PIX firewall supports multimedia applications. It understands that these applications dynamically open various ports and the PIX will not block what it perceives to be normal behavior for this application

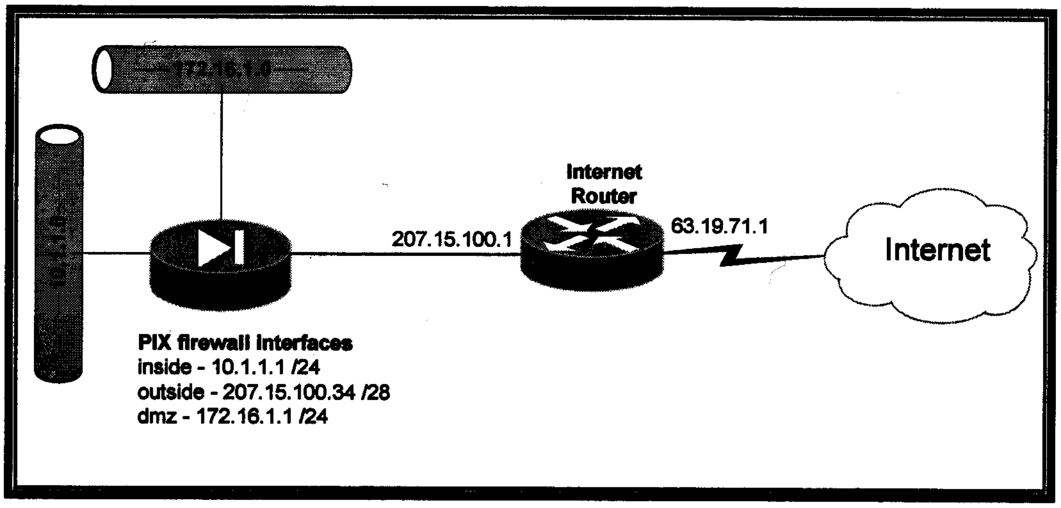
BASIC PIX FIREWALL CONFIGURATION

MINIMUM PIX CONFIGURATION

Cisco identifies six commands needed to make a PIX operational. The following information is used along with Figure 15-1 to describe how we are configuring the PIX.

- The outside interface will be 207.15.100.34 255.255.255.128
- The inside interface will be 10.1.1.1 255.255.255.0
- The dmz interface will be 172.16.1.1 255.255.255.0 with a security level of 50.
- Configure NAT for all inside hosts on 10.1.1.0 255.255.255.0 using a global pool from 207.15.100.35 – 207.15.100.125
- Configure PAT for all dmz hosts on 172.16.1.0 255.255.255.0 using a global address of 207.15.100.11
- The default router for the outside network is 207.15.100.1

Figure 16.1 *Basic PIX topology*



Step 1 Name the interfaces and set security levels. By default, the inside and outside interfaces are already configured. So, if you only plan on using the two default interfaces this command will already be in the configuration. For this exercise, we need to set the DMZ interface and security level.

```
pixfirewall(config)# nameif ethernet2 dmz 50
```

Step 2 Enable the interfaces. By default, all interfaces are shutdown. Use the interface command to enable an interface. In our example, we are manually setting the interfaces to 100Mbps full duplex. Alternatively, you can use auto negotiation by replacing the "100full" with "auto."

```
pixfirewall(config)# interface ethernet0 100full
pixfirewall(config)# interface ethernet1 100full
pixfirewall(config)# interface ethernet2 100full
```

Step 3 Configure each interface with an IP address.

```
pixfirewall(config)# ip address inside 10.1.1.1 255.255.255.0
pixfirewall(config)# ip address outside 207.15.100.34 255.255.255.128
pixfirewall(config)# ip address dmz 172.16.1.1 255.255.255.0
```

Step 4 Configure NAT. Unlike a router, the PIX will not pass any traffic from low security to high security interfaces unless NAT is configured! There are essentially two parts to configure NAT. The first part assigns a NAT ID and the IP addresses subject to the NAT process. The NAT ID is then mapped to the global address.

Do not use `nat <interface> 0`. NAT 0 excludes networks from the NAT process. It is used if you have registered IP addresses that you do not want to NAT. It is also used for VPN configurations as explained in *Section III: Virtual Private Networks.*

```
pixfirewall(config)# nat (inside) 1 10.1.1.0 255.255.255.0 0 0
pixfirewall(config)# nat (dmz) 2 172.16.1.0 255.255.255.0 0 0
```

Note the two 0's that follow the subnet mask. The first number is the maximum number of TCP connections allowed for this particular NAT process. The second number is the embryonic connection limit. An embryonic connection is also known as a TCP half-open. Too many embryonic connections can potentially cause a denial of service attack. The value of 0 for both of these numbers means that the connections are unlimited.

Step 5 Map the inside addresses allowed for NAT to the outside address(es). Configure the PAT address to be used for the pool of addresses to be translated. If an IP address matches NAT ID 1, it will be given an outside IP address from 207.15.100.35 through 207.15.100.125. If an IP address matches NAT ID 2, it will be given the outside IP address of 207.15.100.11.

```
pixfirewall(config)# global (outside) 1 207.15.100.35-207.15.100.125
255.255.255.128
pixfirewall(config)# global (outside) 2 207.15.100.11 255.255.255.128
```

Note that if PAT or NAT has already been configured and these commands are being used to change an existing translation you will need to enter the *clear xlate* command.

Step 6 Configure the route for the outside and inside as necessary. Typically, a default route is needed for the outside. The inside may need a route if there are more networks than the one the PIX is physically attached to. DO NOT configure a default route on each interface.

```
pixfirewall(config)# route outside 0.0.0.0 0.0.0.0 207.15.100.1
```

EXAMPLE - FINAL CONFIGURATION

The highlighted commands illustrate the 6 steps we just completed.

```
pixfirewall# write terminal
Building configuration...
: Saved
:
PIX Version 6.0(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security50
nameif ethernet3 intf3 security15
nameif ethernet4 intf4 security20
nameif ethernet5 intf5 security25
enable password xr4rds12938eT encrypted
passwd Ju12309sjsUxlug encrypted
hostname pixfirewall
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
names
 pager lines 24
no logging timestamp
no logging standby
no logging console
no logging monitor
no logging buffered
no logging trap
logging facility 20
logging queue 512
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 auto shutdown
interface ethernet4 auto shutdown
interface ethernet5 auto shutdown
mtu outside 1500
mtu inside 1500
mtu intf2 1500
mtu intf3 1500
mtu intf4 1500
mtu intf5 1500
ip address inside 10.1.1.1 255.255.255.0
ip address outside 207.15.100.34 255.255.255.128
ip address dmz 172.16.1.1 255.255.255.0
ip address intf3 127.0.0.1 255.255.255.255
ip address intf4 127.0.0.1 255.255.255.255
ip address intf5 127.0.0.1 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
failover ip address intf3 0.0.0.0
failover ip address intf4 0.0.0.0
failover ip address intf5 0.0.0.0
pdm history enable
arp timeout 14400
global (outside) 1 207.15.100.10 255.255.255.128
global (outside) 2 207.15.100.11 255.255.255.128
nat (inside) 1 10.1.1.0 255.255.255.0 0 0
nat (dmz) 2 172.16.1.0 255.255.255.0 0 0
route outside 0.0.0.0 0.0.0.0 207.15.100.1 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
    sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
no snmp-server location
no snmp-server contact
snmp-server community public
no snmp-server enable traps
no floodguard enable
telnet 192.168.10.0 255.255.255.0 inside
telnet timeout 5
terminal width 80
Cryptochecksum:2d4cf99d63effd4277257be78a2b31a8
: end
[OK]
```

PERMITTING TRAFFIC THROUGH THE PIX

By default, the PIX does not pass any traffic from one interface to the other. To allow traffic to originate from a higher security interface with a destination on a lower security interface (i.e. inside to outside) create a NAT and global pair. This concept was briefly covered in the previous section. This section explains how to allow traffic from a source address on a lower security interface with a destination address on a higher security interface. In order for the PIX to permit this traffic through the ASA, a static NAT and either a conduit or access-list must be created.

STATIC TRANSLATIONS

Certain hosts need to have the same global address all the time. Web servers, DNS servers, Mail servers, etc all typically need to have an IP address that never changes. Static translations map a single inside IP address to a single outside IP address. Static addresses take precedence over NAT and global pairs. In other words, a web server that initiates a connection to a host on the Internet will not use the global NAT configured for the inside hosts. It will use the specific static translation assigned to it. See the example below for a web server that is assigned a public address of 207.15.100.150 and an inside address of 10.1.1.50.

```
pixfirewall(config)# static (inside, outside) 207.15.100.50 10.1.1.50
```

CONDUIT COMMAND

The conduit command works in conjunction with a static NAT. Once a static NAT is created, the global IP address is mapped to an inside IP address. However, the PIX will still not pass traffic from the outside interface to the inside. An exception must be created in the ASA. One way to do this is the conduit command. The following is an example of a conduit that permits traffic from any source to the global IP address of 63.95.71.20 with a destination port of 25. This address is for a mail server that should accept connections from any host on the Internet as long as they are sending SMTP e-mail (TCP port 25).

```
pixfirewall(config)# conduit permit tcp host 63.95.71.20 eq smtp any
```

ACCESS-LIST AND ACCESS-GROUP COMMANDS

Cisco is encouraging its customers to transition to access-list instead of conduits. Access lists became available in PIX version 5.0(1). The syntax is very similar to routers except the mask is not a wildcard mask. It is a regular subnet mask just like the mask used on an interface configuration. The following is an example of an access-list.

```
pixfirewall(config)# access-list INBOUND permit tcp any host 208.185.211.71 eq
www
pixfirewall(config)# access-list INBOUND permit tcp 63.1.17.0 255.255.255.0
host 208.15.2.7 eq 23
pixfirewall(config)# access-list INBOUND permit udp any host 208.185.211.75 eq
dns
```

Unlike conduits, the access-lists do not allow traffic through the ASA once they are configured. The access-lists must be applied to an interface using the access-group command.

The access-group command functions the same way as on a router, but the syntax is slightly different. Because the PIX does not have a separate interface configuration, the access-group command must specify which interface the access-list should be applied to. We apply the access-list called "INBOUND" in (incoming connections) on the outside interface. The name INBOUND is arbitrary.

```
pixfirewall(config)# access-group INBOUND in interface outside
```

Access lists can be used to restrict inbound as well as outbound traffic. The outbound command should be replaced with access lists.

Do not configure access lists and conduits on the same PIX firewall!

Advanced Features and Commands

The PIX has many advanced features to accommodate various protocols and block potentially harmful traffic. The PIX has special commands for SMTP, DNS, Java applets, ActiveX, URL filtering, DHCP, multimedia support, virus scanning, etc. Some of these features are automatically part of the PIX ASA, some require manual configuration, and others even require additional software running on external servers.

Fixup Command

The fixup protocol is used by the PIX so it knows how specific protocols behave. Different protocols have special characteristics that the PIX needs to understand in order to allow the return traffic through the ASA. For example, FTP connections open a separate data channel that uses different ports from the initial connection. If the PIX was not aware of this behavior it would deny that return traffic on the separate data channel. But, since the PIX software expects this behavior, it allows the traffic to pass. There are several fixup commands enabled by default on the PIX. The following is a list for version 6.2(2).

```
fixup protocol ftp 21
fixup protocol http 80
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol ils 389
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000
```

Typically, the default protocols are sufficient. However, there may be instances that additional fixup commands may be necessary. Using HTTP on other ports is very common. Frequently, web developers will use port 8080 for development sites that are under construction. Also, SQLNET often uses port 1521 and 1523.

```
pixfirewall(config)# fixup protocol http 8080
pixfirewall(config)# fixup protocol sqlnet 1523
```

Java Applet Filtering

Java applets can be used by an attacker to compromise an internal host. To combat this problem, the PIX firewall can block outgoing Java applets. Java applets need a client host on the inside to contact a server on the Internet that uses Java applets. If a client host downloads malicious Java applet code, it can be used by the attacker to compromise that host. To prevent Java applets from being downloaded use the "filter" command. The following examples demonstrate the different syntax for the **filter java** command. The first example only blocks Java applets on port 80 coming from a source of 10.1.1.0 and a destination of 192.168.1.0. The second example blocks Java applets from ports 1-65535. With the PIX, you can often replace 0.0.0.0 with a single 0 to represent "any."

```
pixfirewall(config)# filter java 80 10.1.1.0 255.255.255.0 192.168.1.0
255.255.0.0
pixfirewall(config)# filter java 1-65535 0 0 0 0
```

ActiveX Blocking

ActiveX is similar to Java applets in that it allows web pages to be much more dynamic and interactive than a regular web page. It also comes with similar security risks. The syntax for blocking ActiveX is nearly identical to Java applets as seen in the example below.

```
pixfirewall(config)# filter activex 80 10.1.1.0 255.255.255.0 192.168.1.0
255.255.0.0
pixfirewall(config)# filter activex 1-65535 0 0 0 0
```

PIX Failover

In order to implement failover, the PIX's must be identical devices including model, memory, network interface cards, and operating system versions. One PIX is considered the "Active" unit while the other is the "Standby" unit. The active unit performs normal network functions while the standby unit monitors the primary unit, ready to take control should the active unit fail or become unavailable.

Both units share the same configuration and also have the same IP address and MAC address. This prevents end hosts and network devices from manual change when a failure occurs. This is similar to the way HSRP on routers operates. Both PIX interfaces must be on the same LAN or VLAN and be able to communicate with each other.

Configuring Failover

Prior to configuring failover, ensure that the PIX firewalls are identical. Also, do not power on the standby unit until you complete the failover configuration on the primary unit.

Configuring failover only requires two commands. The first command simply enables failover on the PIX. The second command is the IP address used by the standby unit. Make sure all active interfaces are given an IP address.

```
pixfirewall(config)# failover
pixfirewall(config)# failover ip address inside 10.1.1.2
pixfirewall(config)# failover ip address dmz 172.16.1.2
pixfirewall(config)# failover ip address outside 207.15.100.2
```

There are several optional and maintenance commands for failover functionality.

`failover active` – causes a unit to become the active unit
`no failover active` – cause a unit to become the standby unit
`failover reset` – clears the failed state of both units and restarts failover
`failover poll [seconds]` – specifies the failover poll interval (this command is only available in version 5.2 and later). The default is 15 seconds.

Stateful Failover

Stateful failover allows the standby unit to keep track of existing connections on the active unit. Should a failure occur, the standby unit will already know about active sessions. This feature first became available in version 5.0.

A dedicated 100 Mbps full duplex connection must be configured on each unit must be configured for stateful failover to work properly. These interfaces must be on the same network. The PIX utilizes a Logical Update (LU), a software module that provides the transport service for stateful failover. The LU contains information about translation tables (xlate, static, and dynamic) and connection (conn) records. These LU's are sent very frequently since each time there is a state change on the active unit, an LU is sent to the standby unit to update its tables. TCP state tables, except for HTTP connections, are transferred. UDP state tables are not transferred except those used to support multi-channel protocols such as H.323.

Applications and sessions that timeout before the failover sequence is complete will have to be reestablished. HTTP and most UDP sessions will also need to be reestablished since their states are not transferred.

Configuring Stateful Failover

Prior to configuring stateful failover, make sure your dedicated 100Mbps interface is configured and crossover cable is correctly attached. Alternatively, you can use a dedicated switch or VLAN and use a regular Cat5 patch cable. Do not use a VLAN that has existing traffic. The name "sfailover" in the example below is arbitrary.

```
pixfirewall(config)# nameif ethernet2 sfailover security90
pixfirewall(config)# failover link sfailover
```

Alias command

The alias command is needed when there is an internal server (typically a web server) that an internal client connects to using the Fully Qualified Domain Name. This is a problem because the external DNS server gives the public IP address for the web server. The alias command instructs the PIX to change the DNS response packet from the public address to the internal IP address. The PIX cannot route traffic out the same interface it was received which is why the alias command is necessary.

Figure 16.2 *Alias command for DNS doctoring*

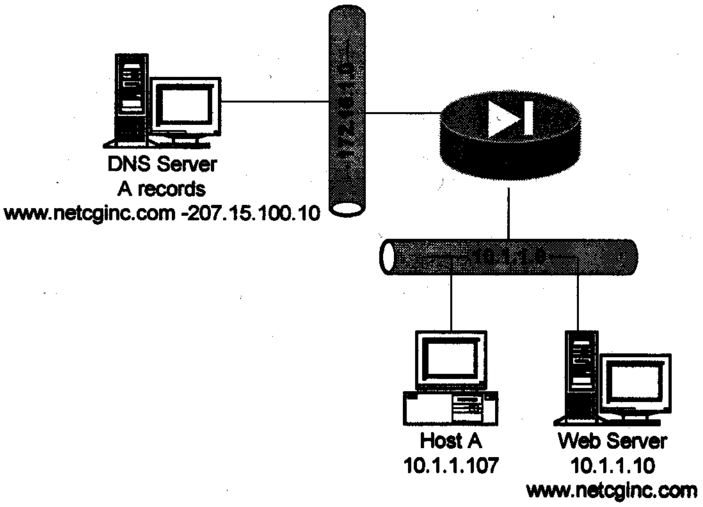
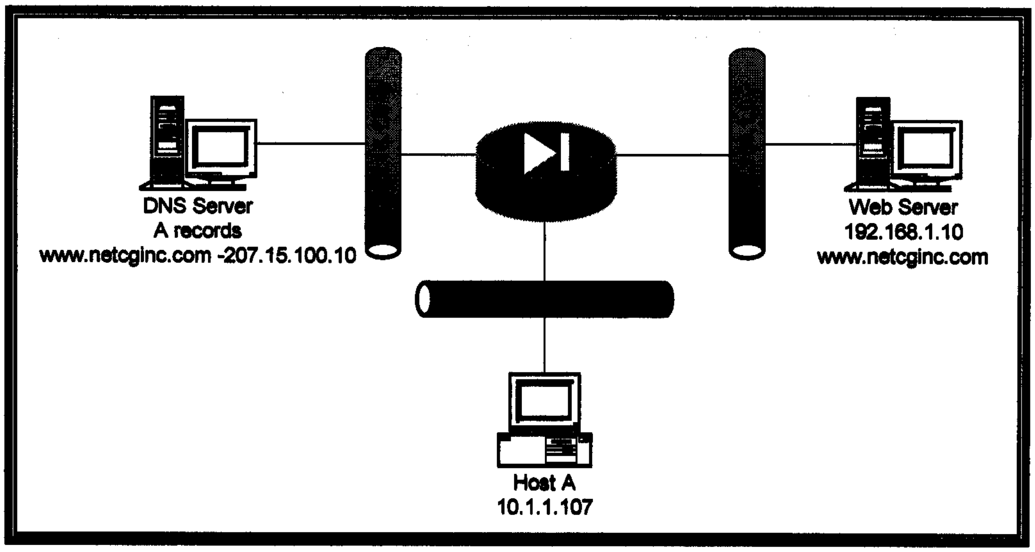


Figure 16.2 shows an example of when the alias command would be needed. Assuming the PIX is already configured for basic connectivity as well as the static NAT for the web server only one command is needed.

```
pix(config)# alias (inside) 10.1.1.10 207.15.100.10 255.255.255.255
```

If the server was on a DMZ instead of the inside, we still have a problem with redirection. In this next example, we want the PIX to perform what is called destination NAT. Destination NAT will map the global address to the address on the DMZ. This allows the inside hosts to communicate with the webserver even though they are given the 207.15.100.10 address by the DNS server.

Figure 16.3 *Alias command for DNAT*



```
pix(config)# alias (inside) 207.15.100.10 192.168.1.10 255.255.255.255
```

DHCP Server

The PIX can be configured as a DHCP server to provide clients with an IP address, DNS server, WINS server, and domain name. DHCP server is available in version 5.2. Prior to version 5.3, 10 DHCP clients were supported by the DHCP server. Version 5.3 supports 25 DHCP clients.

Configuring DHCP has two required and several optional commands.

Step 1 Configure the pool of addresses to be issued.

```
pixfirewall(config)# dhcpd address 10.1.1.100-10.1.1.200 inside
```

Step 2 Enable DHCP on the desired interface. In version 5.2, the only interface that supports DHCP server is the inside interface.

```
pixfirewall(config)# dhcpd enable inside
```

Step 3 Configure any optional commands such as DNS, WINS, domain, or lease time. If you have additional settings to provide with the DHCP reply, enter them prior to enabling DHCP on the interface. The default lease is 3,600 seconds (1 hour). We set our lease to one day.

```
pixfirewall(config)# dhcpd dns 198.6.1.3
pixfirewall(config)# dhcpd wins 10.1.1.99
pixfirewall(config)# dhcpd domain netcginc.com
pixfirewall(config)# dhcpd lease 86400
```

DHCP Client

DHCP client is used when a static IP address is not provided by the ISP. Typically, DHCP client is used in DSL and Cable networks that only provide an IP address and default route via DHCP. The DHCP provided IP address can also be used for PAT so that you can use your single IP address to route traffic for many inside hosts. DHCP client does not work with a failover configuration. The "setroute" option allows the PIX to accept a default route via DHCP.

```
pixfirewall(config)# ip address outside dhcp setroute
```

Outbound and Apply

By default, all traffic from a higher security interface to a lower security interface is allowed. The outbound and apply commands are used to restrict access from a higher security interface to a lower security interface. Outbound access can be restricted by source address and port or by destination address and port. The command **outgoing_src** filters outbound traffic based on the source address. The command **outgoing_dest** filters outbound traffic based on the destination address. The **outgoing_src** and **outgoing_dest** are filtered independently. If a port is not specified it defaults to TCP. The apply command applies the outbound filter to an interface.

Cisco recommends that its customers transition from using the outbound and apply commands and use access lists for all PIX filtering.

This first example permits users on the 10.1.1.0 /24 network to access the Internet, but denies the 10.2.2.0 /24 network. After the mask is the port and protocol option. If no port or protocol is specified, it will default to IP although the command reference says it will default to TCP. This may be version dependent as all of our tests were conducted on PIX version 6.1(1). However, the command reference for 6.1(1) indicates the default protocol is TCP. This should prove that the documentation is not necessarily accurate so make sure you understand the true behavior of the devices you are configuring. Also, make sure your "list_id" numbers from the outbound and apply match. In this example, we are using a "list_id" of 1.

```
pixfirewall(config)# outbound 1 permit 10.1.1.0 255.255.255.0 0 ip
pixfirewall(config)# outbound 1 deny 10.2.2.0 255.255.255.0 0 ip
pixfirewall(config)# apply (inside) 1 outgoing_src
```

The next example denies users from accessing Hotmail because of excessive traffic and the ability to download email attachments without coming through the corporate email system for antivirus scanning.

```
pixfirewall(config)# outbound 2 deny 64.4.43.7 255.255.255.255 0 ip
pixfirewall(config)# apply (inside) 2 outgoing_dest
```

RIP

The only dynamic routing protocol the PIX can support is RIP. It can be configured for RIP version 1 or 2. However, the PIX does not pass RIP updates from one interface to another. If routers need to send routing updates through the PIX, a GRE tunnel will be necessary.

RIP can run in either passive or default mode. Passive mode listens for routes and places them in its routing table. Default mode will broadcast a default route in to the RIP domain on that interface.

Using RIP version 2, the behavior changes beginning with version 5.3. Older versions send and receive RIP using the 255.255.255.255 broadcast address. Newer versions use the 224.0.0.9 broadcast address. However, only Gigabit and Fast Ethernet interfaces support the multicast address.

CONFIGURING RIP PASSIVE

With RIP Passive enabled, the PIX receives routes from the other RIP speaking devices on the interface configured for RIP. The PIX does not advertise any of its interfaces or networks, but it does know how to reach the rest of the internal network.

```
pixfirewall(config)# rip inside passive
```

CONFIGURING RIP DEFAULT

With RIP Default enabled, the PIX advertises a default route to the other RIP speaking routers on the interface configured for RIP. This default route will propagate throughout the RIP network so the rest of the RIP domain knows the default route out of the network.

```
pixfirewall(config)# rip inside default
```

CONFIGURING RIP VERSION 2 WITH AUTHENTICATION

RIP version 2 allows for authentication and use the use of multicast address instead of broadcasting updates. Make sure your PIX is running version 5.3 or later since earlier versions did not support the multicast address. We configured the PIX and R14 for RIP version 2 with authentication in the following two examples. R14 has two networks configured for RIP (10.0.0.0 and 192.168.13.0).

```
pixfirewall(config)# rip inside default version 2 authentication md5 cisco 1

r14# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.1.1.1 to network 0.0.0.0

C    192.168.13.0/24 is directly connected, ATM2/0
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, Ethernet0/0
R*   0.0.0.0/0 [120/1] via 10.1.1.1, 00:00:17, Ethernet0/0
```

-OR-

```
pixfirewall(config)# rip inside passive version 2 authentication md5 cisco 1

pixfirewall# show route
        inside 10.1.1.0 255.255.255.0 10.1.1.1 1 CONNECT static
        sfailover 172.16.1.0 255.255.255.0 172.16.1.1 1 CONNECT static
        inside 192.168.13.0 255.255.255.0 10.1.1.14 1 RIP
```

SYSOPT COMMAND

The **sysopt** command allows the PIX to be configured and tuned for the various security features. It also allows for the PIX to permit traffic terminated on the PIX itself thereby bypassing the ASA and configured access lists or conduits. Allow of these commands are disabled by default unless otherwise specified.

sysopt connection permit-ipsec – allows IPSec traffic to bypass conduit or access-list command statement checking.

sysopt connection permit-l2tp - allows L2TP traffic to bypass conduit or access-list command statement checking.

sysopt connection permit-pptp - allows PPTP traffic to bypass conduit or access-list command statement checking.

sysopt connection timewait - forces each TCP connection to linger in a shortened TIME_WAIT state of at least 15 seconds after the final normal TCP close-down sequence. A normal TCP close sequence has one host initiate the closing sequence and the other host acknowledges the close sequence. A simultaneous TCP close may cause performance problems for some hosts such as some WinSock mainframe clients and older versions of HP-UX. One of the hosts may linger in the CLOSING state. If you have such hosts, this command will enable a quiet time window for the abnormal close down sequence to complete.

sysopt connection tcpmss *<bytes>* - forces TCP proxy connections to have a maximum segment size no greater than the value configured (in bytes). The default value is 1380 bytes. This command is recommended in a network being attacked being with overly aggressive TCP or

HTTP stacks and a faulty path MTU value that is degrading the performance of the PIX. This command is enabled by default.

`sysopt noproxyarp <if_name>` - disable proxy-arps on a PIX Firewall interface.

`sysopt nodnsalias inbound` – disables inbound embedded DNS Address record fixups according to aliases that apply to the A record address.

`sysopt nodnsalias outbound` – disables outbound DNS Address record replies.

`sysopt security fragguard` - enables the IP Frag Guard feature. This feature prevent non-initial IP fragments from passing through that PIX and it limits the number of valid fragments to each host at 100 per second. This helps prevent fragment style attacks such as teardrop and land.c. The IP Frag Guard feature operates on all interfaces in the PIX Firewall and cannot be selectively enabled or disabled by interface. However, enabling this command may break valid traffic if the packets arrive out of order. Certain versions of Linux sent fragment packets in reverse order which would automatically be denied by the PIX. Use this command with caution.

`sysopt radius ignore-secret` - ignores authenticator key to avoid retransmit. Some RADIUS servers do not include the key in the authenticator hash in the accounting acknowledgment response. This can cause the PIX to continually retransmit the accounting request. This command will tell the PIX to ignore the key in the authenticator of accounting acknowledgement.

`sysopt uauth allow-http-cache` – allows the web browser to supply a username and password from its cache for AAA authentication. If this command is not enabled, the PIX will require the user to authentice each time the uauth timer expires.

`sysopt route dnat` - specifies that when an incoming packet does a route lookup, the incoming interface is used to determine which interface the packet should go to, and which is the next hop.

MAIL GUARD

The PIX uses the Mail Guard feature to allow outside email to relay mail to the inside SMTP server with an additional level of security. The PIX only permits SMTP mail on TCP port 25. It also only allows specific SMTP commands for added security. These commands are defined in RFC 821 section 4.1.1.

- MAIL
- HELO
- RCPT
- DATA
- RSET
- NOOP
- QUIT

DNS GUARD

When a host on the inside network sends a DNS resolve request, the PIX opens up a translation slot and will permit the return traffic. Hosts may send queries to multiple DNS servers. So, the PIX may see multiple responses. However, it will only permit the first response through the ASA and then will immediately close the UDP connection. It will not permit any additional responses nor will it wait for the UDP timer (default is 2 minutes) to expire.

SERVICE RESET INBOUND

With the command **service resetinbound** enabled, the PIX will issue a TCP reset (RST) to inbound TCP connections that are denied by access-lists or conduits. It will also issue a RST if user authorization fails. If this option is not enabled, the PIX drops the packet without sending anything back to the source.

This command is particularly useful to prevent slow email service. Some mail servers rely on the IDENT protocol. The problem is when an outside host is trying to send mail to the inside mail server, it will wait for the IDENT timeout which may take a minute or two. With **service resetinbound**, the PIX will notify the source host to reset the connection and it will no longer wait for the IDENT timeout.

TYPICAL GOTCHAS!

- Leaving interfaces shutdown
- Missing a NAT or Global statement
- Missing static, default, or RIP routes for simple connectivity
- Deleting an ACL and forgetting to reapply the access-group

Section IV

Virtual Private Networks

CHAPTER 17

VPN OVERVIEW

Cisco VPNsare built using several different products including the PIX firewall, Cisco routers, and VPN 3000/5000 Concentrators. Cisco's VPN implementation can make use of several protocols including Data Encryption Standard (DES), Triple DES (3DES), IP Secure (IPSec), and Internet Key Exchange (IKE). Understanding how each of these components fits in to a VPN is integral to being able to configure Cisco VPN solutions

IP Secure (IPSec)

There are four parts to an IPSec VPN: Encryption, Security Associations (key exchange), Data Integrity, and Origin Authentication. Encryption is provided by using either DES or 3DES. Security Associations are typically handled by Internet Key Exchange (IKE) which can use pre-shared keys, RSA encryption, or RSA signatures with digital certificates. Security Associations can be configured manually without IKE, but are not frequently configured with Cisco VPN's. Data Integrity ensures that the data has not changed in transit from source to destination. Data integrity is handled by hashing algorithms such as Encapsulating Security Payload (ESP) or Authentication Header (AH) along with the corresponding integrity-checking hash algorithms Message Digest 5 (MD5) or Secure Hash Algorithm (SHA-1). Origin Authentication is provided by using Digital Signatures or Digital Certificates. Origin Authentication is the only part of a Cisco IPSec VPN that is optional.

Data Encryption Standard (DES)

Data Encryption Standard (DES) is a widely used method of data encryption using a private key. Initially the U.S. government restricted export to other countries because at the time they believe it was extremely difficult to break. There are 72,000,000,000,000,000 or more possible encryption keys available. Each message has a new key chosen at random. Both the sender and the receiver must know and use the same private key.

DES applies a 56-bit key to each 64-bit block of data. The process can run in several modes and involves 16 rounds or operations. This is considered "weak" encryption, therefore many companies use Triple DES, which applies three keys in succession. It is recommended that companies transition to 3DES because DES can be broken or cracked significantly easier than 3DES. DES has already been cracked several times. The results have been made public on the Internet and in books. The methods to crack DES are well documented if the attacker has the right hardware and software. For the curious, O'Reilly's book titled "Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design" explains in detail how DES can be cracked.

TRIPLE DES (3DES)

Triple DES (3DES) is another common form of encryption that works the same as DES except that 3DES applies three keys in succession, creating a 168-bit key that is extremely difficult to break. The U.S. government still does not allow 3DES software to be exported outside the U.S. and Canada.

INTERNET KEY EXCHANGE (IKE)

IKE supports 3 different types of authentication methods: pre-shared keys, RSA encryption, and RSA signatures with digital certificates. IKE, a hybrid protocol, is responsible for key exchange between two VPN peers. It implements the security protocols Oakley Key Exchange and Skeme Key Exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework.

Pre-shared keys are generally used in small VPN implementations due to the requirement to manually configure a key for each peer on a VPN device. This does not scale well in large VPN implementations.

Both RSA types rely on public key cryptography. RSA without digital certificates does not provide nonrepudiation. Nonrepudiation allows a third party to be able to prove that a communication between two other parties occurred. This is desirable if there is a need to prove certain communications took place. Repudiation is the opposite and is used if you do not want a trace of your communications.

Typical Gotchas!

- Hardware does not support IPSec or the components required (i.e. router supports IPSec but not 3DES)
- Misunderstanding of the components asked for

CHAPTER 18

VPN CONFIGURATION

The following are five different examples of setting up Cisco VPNs using various products and protocols. These examples are key to understanding the complex configuration of a Cisco VPN.

Example 1 – Router to router VPN with 3DES and IKE pre-shared keys
Example 2 – PIX to PIX VPN with DES, 3DES and IKE pre-shared keys
Example 3 – PIX to 2 remote routers with 3DES and IKE pre-shared keys
Example 4 – PIX to router with 3DES, IKE pre-shared keys and NAT
Example 5 – PIX to router with DES and Manual keys

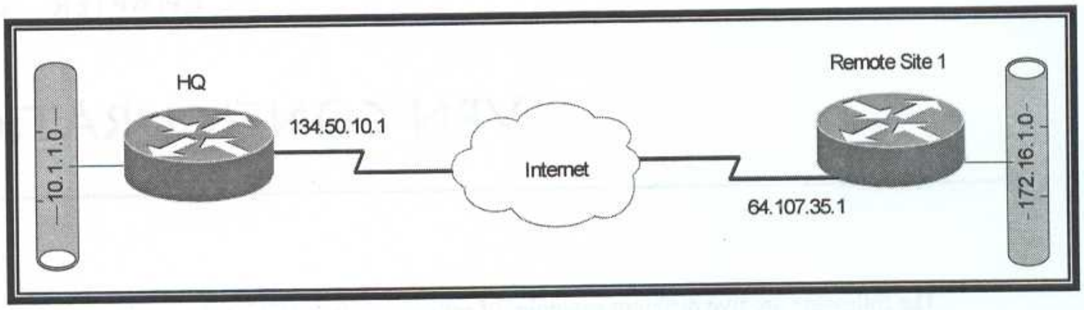
The steps listed in this configuration need not be performed in this order. However, you should understand how each command relates to each other before changing this order. It is recommended to have console access to any device you are configuring for a VPN. If you make certain mistakes such as applying a crypto map to an interface prior to configuring an access-list, IPSec will force a PIX to encrypt ALL traffic thereby stopping all outbound traffic.

These configurations require either DES or 3DES capability depending on the example. DES versions of software for the PIX firewall can be requested free from Cisco, however 3DES versions require additional licensing fees. Downloading most software images from Cisco will require a valid support contract with software support. Before beginning any configurations on your equipment make sure you have enough system resources (processor, memory, flash) as well as an image capable of performing the VPN functions you are deploying.

Example 1 - Router to Router VPN using 3DES and IKE pre-shared keys

This example demonstrates two routers configured for a VPN tunnel over the Internet. The Headquarters site's LAN uses the network is 10.1.1.0. The headquarters public IP address is 134.50.10.1. The remote site's network is 172.16.1.0. The remote site's public IP address is 64.107.35.1. Configure 3DES and IKE pre-shared keys.

Figure 18.1 *Router to router VPN connection*



Headquarters Router Configuration

Step 1 Specify a hostname.

```
router(config)# hostname hq-vpn-rtr
```

Step 2 Configure the ISAKMP key and address of the remote VPN router or PIX. The key exchange portion (ISAKMP commands) is known as Phase I. Phase I handles peer establishment by creating a secure authenticated tunnel after the key exchange is successful.

```
hq-vpn-rtr (config)# crypto isakmp key ciscovpnkey address 64.107.35.1
```

Step 3 Configure ISAKMP policy. This is where you configure encryption, authentication, hash, and lifetime values. Each of these policies already has a default and only needs a configuration command if you want to use something other than the default. To see the default settings type the global command **show crypto isakmp policy**. In our example, we are leaving the hash, Diffie-Hellman group, and lifetimes at their default of SHA-1, Group 1, and 86400 seconds.

```
hq-vpn-rtr(config)# crypto isakmp policy 10
hq-vpn-rtr(config-isakmp)# encryption 3des
hq-vpn-rtr(config-isakmp)# authentication pre-share
```

Step 4 Configure an access-list to specify which traffic (source to destination) will be encrypted. All packets with a source address of 10.1.1.0 and a destination of 172.16.1.0 will be encrypted in this example.

The access-list configuration marks the beginning of Phase II, which includes all of the IPSec commands (access-list, transform set, and crypto map commands). Phase II peers use the authenticated and secure channel to negotiate services for IPSec. Remember that it is not required to configure these commands in this exact order.

```
hq-vpn-rtr(config)# access-list 100 permit ip 10.1.1.0 0.0.0.255
172.16.1.0 0.0.0.255
```

Step 5 Configure the transform-set. This is where you set the IPSec encryption type and hash combination. There are many choices for IPSec encryption (DES, 3DES, or null), security protocol (ESP and/or AH), and hash type (SHA-1 or MD5). You can set the encryption type (cipher) first or you can set the hash/security protocol first. Note that the word "strong" is arbitrary and can be any word you choose to define your transform set. We typically use "strong" for 3DES transform sets and "weak" for DES transform sets.

```
hq-vpn-rtr(config)# crypto ipsec transform-set strong ?
ah-md5-hmac    AH-HMAC-MD5 transform
ah-sha-hmac    AH-HMAC-SHA transform
comp-lzs       IP Compression using the LZS compression algorithm
esp-3des       ESP transform using 3DES(EDE) cipher (168 bits)
esp-des        ESP transform using DES cipher (56 bits)
esp-md5-hmac   ESP transform using HMAC-MD5 auth
esp-null       ESP transform w/o cipher
esp-sha-hmac   ESP transform using HMAC-SHA auth
```

In this example, we have selected esp-3des as the cipher and ESP as the security protocol. Now we can see that we must choose our hash type.

```
hq-vpn-rtr(config)# crypto ipsec transform-set strong esp-3des ?
ah-md5-hmac    AH-HMAC-MD5 transform
ah-sha-hmac    AH-HMAC-SHA transform
comp-lzs       IP Compression using the LZS compression algorithm
esp-md5-hmac   ESP transform using HMAC-MD5 auth
esp-sha-hmac   ESP transform using HMAC-SHA auth
```

Here is the complete transform set.

```
hq-vpn-rtr(config)# crypto ipsec transform-set strong esp-3des esp-sha-
hmac
```

Step 6 Configure the crypto map. This is where you set the IPSec peer IP address of the other router you are connecting to. Since a router can be configured for multiple transform sets this is the section where you tell this particular crypto map which transform set to use. Lastly, this is where you specify the traffic to encrypt using the `match address <access-list number>` command.

```
hq-vpn-rtr(config)# crypto map netcg 10 ipsec-isakmp
hq-vpn-rtr(config-crypto-map)# set peer 64.107.35.1
hq-vpn-rtr(config-crypto-map)# set transform-set strong
hq-vpn-rtr(config-crypto-map)# match address 100
```

Step 7 Configure the inside interface of the router. Interface descriptions are optional but are a highly recommended design practice.

```
hq-vpn-rtr(config)# interface FastEthernet0/1
hq-vpn-rtr(config-if)# ip address 10.1.1.1 255.255.255.0
hq-vpn-rtr(config-if)# description Inside Network
```

Step 8 Configure the outside interface of the router. The crypto map should be applied to the outside interface.

```
hq-vpn-rtr(config)# interface Serial0/0
hq-vpn-rtr(config-if)# description Internet Connection
hq-vpn-rtr(config-if)# ip address 134.50.10.1 255.255.255.252
hq-vpn-rtr(config-if)# crypto map netcg
```

REMOTE SITE1 ROUTER CONFIGURATION

Here is the other end of the VPN tunnel. All the configuration parameters should remain the same except for IP addresses and the access-list(s).

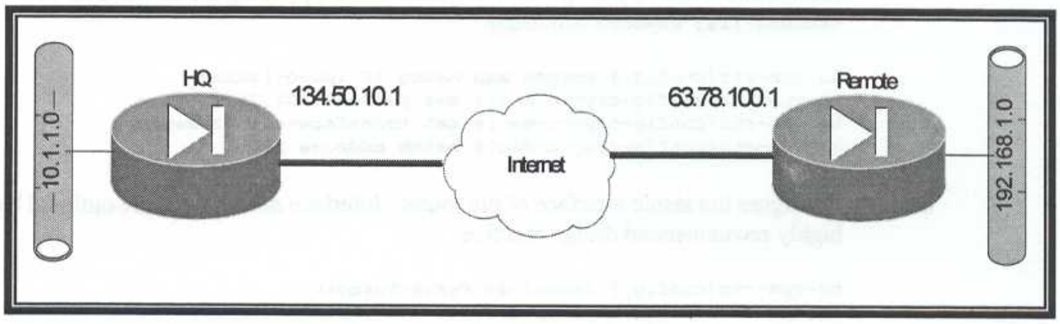
```
router(config)# hostname site1-rtr-vpn
site1-rtr-vpn(config)# crypto isakmp key ciscovpnkey address 134.50.10.1
site1-rtr-vpn(config)# crypto iskakmp policy 10
site1-rtr-vpn(config-isakmp)# encryption 3des
site1-rtr-vpn(config-isakmp)# authentication pre-share
```

```
site1-rtr-vpn(config)# access-list 100 permit ip 172.16.1.0 0.0.0.255 10.1.1.0
0.0.0.255
site1-rtr-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-hmac
site1-rtr-vpn(config)# crypto map netcg 10 ipsec-isakmp
site1-rtr-vpn(config-crypto-map)# set peer 134.50.10.1
site1-rtr-vpn(config-crypto-map)# set transform-set netcg
site1-rtr-vpn(config-crypto-map)# match address 100
site1-rtr-vpn(config)# interface FastEthernet0/1
site1-rtr-vpn(config-if)# ip address 172.16.1.1 255.255.255.0
site1-rtr-vpn(config-if)# description Inside Network
site1-rtr-vpn(config)# interface Serial0/0
site1-rtr-vpn(config-if)# description Internet Connection
site1-rtr-vpn(config-if)# ip address 64.107.35.1 255.255.255.252
site1-rtr-vpn(config-if)# crypto map netcg
```

Example 2 - PIX to PIX VPN with 3DES and IKE pre-shared keys

This example demonstrates two PIX firewalls configured for a VPN tunnel over the Internet. The Headquarters site's LAN uses the network is 10.1.1.0. The headquarters public IP address is 134.50.10.1. The remote site's network is 192.168.1.0. The remote site's public IP address is 63.78.100.1. Configure 3DES and IKE pre-shared keys.

Figure 18.2 *PIX to PIX connection*



Headquarters Site PIX Configuration

Although the PIX commands are very similar to a router, there are some subtle differences.

Step 1 Configure interfaces for outside and inside. Security0 is always used for the outside interface. Security100 is always used for the inside interface. Other interfaces, such as those you would use for a DMZ, use values between 1-99.

```
hq-pix-vpn(config)# nameif ethernet 0 outside security0
hq-pix-vpn(config)# nameif ethernet1 inside security100

hq-pix-vpn(config)# ip address outside 134.50.10.1 255.255.255.0
hq-pix-vpn(config)# ip address inside 10.1.1.4 255.255.255.0
```

Step 2 Configure the PIX to allow IPSec connections to the firewall itself. Without this command, the firewall will not accept IPSec connections unless there are specific access-lists or conduits configured. Sysopt commands are used to bypass conduit or access-list permit commands. All packets that arrive on the VPN tunnel will not be checked by the conduits or access-lists configured on the PIX.

```
hq-pix-vpn(config)# sysopt connection permit-ipsec
```

Step 3 Configure the ISAKMP policy.

```
hq-pix-vpn(config)# isakmp key ciscovpnkey2 address 63.78.100.1 netmask
    255.255.255.255

hq-pix-vpn(config)# isakmp policy 10 authentication pre-share
hq-pix-vpn(config)# isakmp policy 10 encryption 3des
hq-pix-vpn(config)# isakmp policy 10 hash sha
hq-pix-vpn(config)# isakmp policy 10 group 1
hq-pix-vpn(config)# isakmp policy 10 lifetime 43200

hq-pix-vpn(config)# isakmp enable outside
```

Step 4 Configure the access-list of the traffic you want to encrypt.

```
hq-pix-vpn(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
```

Step 5 Configure the transform set.

```
hq-pix-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-
hmac
```

Step 6 Configure the crypto map and apply it to the outside interface.

```
hq-pix-vpn(config)# crypto map pixtopixvpn 10 ipsec-isakmp
hq-pix-vpn(config)# crypto map pixtopixvpn 10 match address 100
hq-pix-vpn(config)# crypto map pixtopixvpn 10 set peer 63.78.100.1
hq-pix-vpn(config)# crypto map pixtopixvpn 10 set transform-set strong

hq-pix-vpn(config)# crypto map pixtopixvpn interface outside
```

REMOTE SITE2 ROUTER CONFIGURATION

```
site2-pix-vpn(config)# nameif ethernet 0 outside security0
site2-pix-vpn(config)# nameif ethernet1 inside security100

site2-pix-vpn(config)# ip address outside 63.78.100.1 255.255.255.0
site2-pix-vpn(config)# ip address inside 192.168.1.1 255.255.255.0

site2-pix-vpn(config)# sysopt connection permit-ipsec

site2-pix-vpn(config)# isakmp key ciscovpnkey2 address 134.50.10.1 netmask
255.255.255.255

site2-pix-vpn(config)# isakmp policy 10 authentication pre-share
site2-pix-vpn(config)# isakmp policy 10 encryption 3des
site2-pix-vpn(config)# isakmp policy 10 hash sha
site2-pix-vpn(config)# isakmp policy 10 group 1
site2-pix-vpn(config)# isakmp policy 10 lifetime 43200

site2-pix-vpn(config)# isakmp enable outside

site2-pix-vpn(config)# access-list 100 permit ip 192.168.1.0 255.255.255.0
10.1.1.0 255.255.255.0

site2-pix-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-
hmac

site2-pix-vpn(config)# crypto map pixtopixvpn 10 ipsec-isakmp
```

```
site2-pix-vpn(config)# crypto map pixtopixvpn 10 match address 100
site2-pix-vpn(config)# crypto map pixtopixvpn 10 set peer 134.50.10.1
site2-pix-vpn(config)# crypto map pixtopixvpn 10 set transform-set strong

site2-pix-vpn(config)# crypto map pixtopixvpn interface outside
```

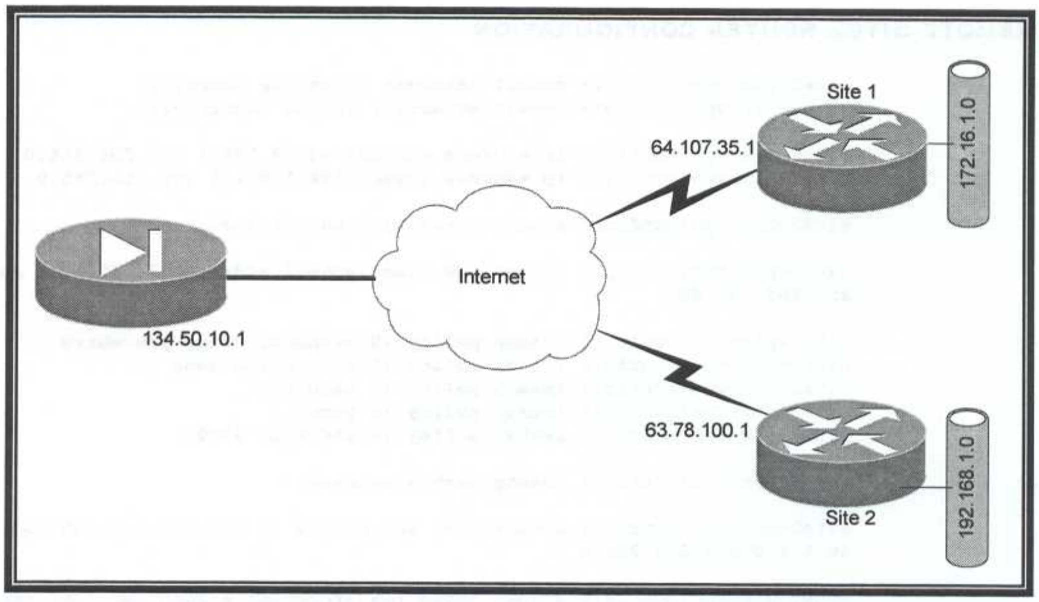
Example 3 – PIX to two remote routers with DES, 3DES, and IKE pre-shared keys

This example builds on the previous two examples. Once again, the headquarters site uses network 10.1.1.0. There are two remote sites using networks 172.16.1.0 (site1) and 192.168.1.0 (site2). The remote routers are configured similar to the previous two examples. Configure the connection to site1 for 3DES encryption and the SHA hash. Configure the connection to site2 for DES and MD5. The PIX at the headquarters need to be configured to connect to each remote site, it will also need to be configured for two different ISAKMP policies and transform sets.

Keep in mind that in a hub and spoke topology the remote sites will not be able to send data to each other without configuring an IPSec tunnel between each remote site. If this is desirable, configure a full mesh topology with each VPN device connecting to each other. This will result in two tunnels on each device instead of one. In large VPN environments, this becomes a scaling problem if there are dozens of VPN connections. With large-scale deployments, it is often recommended to switch to a public key configuration for scalability. See Chapter 18: Certificate Authority for a sample configuration using RSA public keys.

Ensure you create both ISAKMP keys/policies, both transform sets, and use ONLY one crypto map. Multiple crypto map names cannot be applied to an interface. Using multiple numbers in a crypto map as shown below easily solves this problem.

Figure 18.3 *PIX to two remote sites*



HEADQUARTERS PIX CONFIGURATION

```
hq-pix-vpn(config)# nameif ethernet 0 outside security0
hq-pix-vpn(config)# nameif ethernet1 inside security100
hq-pix-vpn(config)# ip address outside 134.50.10.1255.255.255.0
hq-pix-vpn(config)# ip address inside 10.1.1.4 255.255.255.0

hq-pix-vpn(config)# sysopt connection permit-ipsec

hq-pix-vpn(config)# isakmp enable outside
hq-pix-vpn(config)# isakmp key ciscovpnkey1 address 64.107.35.1 netmask
255.255.255.255
hq-pix-vpn(config)# isakmp key ciscovpnkey2 address 63.78.100.1 netmask
255.255.255.255

hq-pix-vpn(config)# isakmp policy 10 authentication pre-share
hq-pix-vpn(config)# isakmp policy 10 encryption 3des
hq-pix-vpn(config)# isakmp policy 10 hash sha
hq-pix-vpn(config)# isakmp policy 10 group 1
hq-pix-vpn(config)# isakmp policy 10 lifetime 43200

hq-pix-vpn(config)# isakmp policy 20 authentication pre-share
hq-pix-vpn(config)# isakmp policy 20 encryption des
hq-pix-vpn(config)# isakmp policy 20 hash md5
hq-pix-vpn(config)# isakmp policy 20 group 1
hq-pix-vpn(config)# isakmp policy 20 lifetime 86400

hq-pix-vpn(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
172.16.1.0 255.255.255.0
hq-pix-vpn(config)# access-list 101 permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0

hq-pix-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-hmac
hq-pix-vpn(config)# crypto ipsec transform-set weak esp-des esp-md5-hmac

hq-pix-vpn(config)# crypto map pixtositevpn 10 ipsec-isakmp
hq-pix-vpn(config)# crypto map pixtositevpn 10 match address 100
hq-pix-vpn(config)# crypto map pixtositevpn 10 set peer 64.107.35.1
hq-pix-vpn(config)# crypto map pixtositevpn 10 set transform-set strong
hq-pix-vpn(config)# crypto map pixtositevpn 20 ipsec-isakmp
hq-pix-vpn(config)# crypto map pixtositevpn 20 match address 101
hq-pix-vpn(config)# crypto map pixtositevpn 20 set peer 63.78.100.1
hq-pix-vpn(config)# crypto map pixtositevpn 20 set transform-set weak
hq-pix-vpn(config)# crypto map pixtositevpn interface outside
```

REMOTE SITE1 ROUTER CONFIGURATION

```
router(config)# hostname site1-vpn-rtr

site1-rtr-vpn(config)# crypto isakmp key ciscovpnkey address 134.50.10.1

site1-rtr-vpn(config)# crypto iskakmp policy 10
site1-rtr-vpn(config-isakmp)# encryption 3des
site1-rtr-vpn(config-isakmp)# authentication pre-share

site1-rtr-vpn(config)# access-list 100 permit ip 172.16.1.0 0.0.0.255 10.1.1.0
0.0.0.255

site1-rtr-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-hmac

site1-rtr-vpn(config)# crypto map netcg 10 ipsec-isakmp
site1-rtr-vpn(config-crypto-map)# set peer 134.50.10.1
site1-rtr-vpn(config-crypto-map)# set transform-set netcg
site1-rtr-vpn(config-crypto-map)# match address 100

site1-rtr-vpn(config)# interface FastEthernet0/1
site1-rtr-vpn(config-if)# ip address 172.16.1.1 255.255.255.0
```

```
site1-rtr-vpn(config-if)# description Inside Network

site1-rtr-vpn(config)# interface Serial0/0
site1-rtr-vpn(config-if)# description Internet Connection
site1-rtr-vpn(config-if)# ip address 64.107.35.1 255.255.255.252
site1-rtr-vpn(config-if)# crypto map netcg
```

Remote Site2 Router Configuration

```
router(config)# hostname site2-vpn-rtr

site2-rtr-vpn(config)# crypto isakmp key ciscovpnkey address 134.50.10.1

site2-rtr-vpn(config)# crypto iskakmp policy 10
site2-rtr-vpn(config-isakmp)# encryption 3des
site2-rtr-vpn(config-isakmp)# authentication pre-share

site2-rtr-vpn(config)# access-list 100 permit ip 192.168.1.0 0.0.0.255
10.1.1.0 0.0.0.255

site2-rtr-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-hmac

site2-rtr-vpn(config)# crypto map netcg 10 ipsec-isakmp
site2-rtr-vpn(config-crypto-map)# set peer 134.50.10.1
site2-rtr-vpn(config-crypto-map)# set transform-set netcg
site2-rtr-vpn(config-crypto-map)# match address 100

site2-rtr-vpn(config)# interface FastEthernet0/1
site2-rtr-vpn(config-if)# ip address 192.168.1.1 255.255.255.0
site2-rtr-vpn(config-if)# description Inside Network

site2-rtr-vpn(config)# interface Serial0/0
site2-rtr-vpn(config-if)# description Internet Connection
site2-rtr-vpn(config-if)# ip address 63.78.100.1 255.255.255.252
site2-rtr-vpn(config-if)# crypto map netcg
```

Example 4 – PIX to Router with 3DES, IKE Pre-Shared Keys and NAT

Typically, IPSec implementations are configured on Internet perimeter routers or firewalls that must also perform NAT. The PIX Firewall and Cisco IOS software require additional commands to get NAT and IPSec to operate concurrently. You will need to enter the following commands in addition to the final configuration from Example 3.

PIX IPSec Configuration with NAT

Step 1 Create the access-list of networks that you do not want to be sent to the NAT process because they need to be encrypted by IPSec. Traffic with a source of 10.1.1.0 and a destination of 10.2.1.0 are not allowed to processed by NAT.

```
pixfirewall(config)# access-list nonat permit ip 10.1.1.0 255.255.255.0
10.2.1.0 255.255.255.0
```

Step 2 Create the NAT pool.

```
pixfirewall(config)# global (outside) 1 216.126.141.39-216.126.141.50
```

Step 3 Configure NAT commands. The command `nat (inside) 0` is actually used for networks that you want to prevent from NAT. As long as packets do not match the `access-list nonat` list, the packets are processed by NAT. The `nat (inside) 1` says that all traffic with a source of 10.1.1.0 will be NAT'd except for traffic excluded by the **nat (inside) 0** command.

```
pixfirewall(config)# nat (inside) 0 access-list nonat
pixfirewall(config)# nat (inside) 1 10.1.1.0 255.255.255.0 0 0
```

Router IPSec Configuration with NAT

Step 1 Create the access-list of networks that you do not want to be sent to the NAT process because they need to be encrypted by IPSec. Traffic with a source address from network 10.2.1.0 and a destination address on 10.1.1.0 is not allowed to be processed by NAT. You should notice that this access-list has a deny statement instead of a permit. On a PIX the **nat (inside) 0** command is actually a list of networks that are blocked from the NAT process. Therefore, the PIX access-list must have a permit.

```
hq-vpn-rtr(config)# access-list 100 deny ip 10.2.1.0 0.0.0.255 10.1.1.0
0.0.0.255
```

Step 2 Create the route-map to be used by the NAT pool.

```
hq-vpn-rtr(config)# route-map NONAT permit 10
hq-vpn-rtr(config-route-map)# match ip address 100
```

Step 3 Create the NAT pool.

```
hq-vpn-rtr(config)# ip nat pool nat-pool 63.78.100.10 63.78.100.250
netmask 255.255.255.0
```

Step 4 Create the source NAT command. This command uses the NAT pool we just created as well as the route-map to block traffic that should be processed by IPSec instead of NAT.

```
hq-vpn-rtr(config)# ip nat inside source route-map NONAT pool nat-pool
```

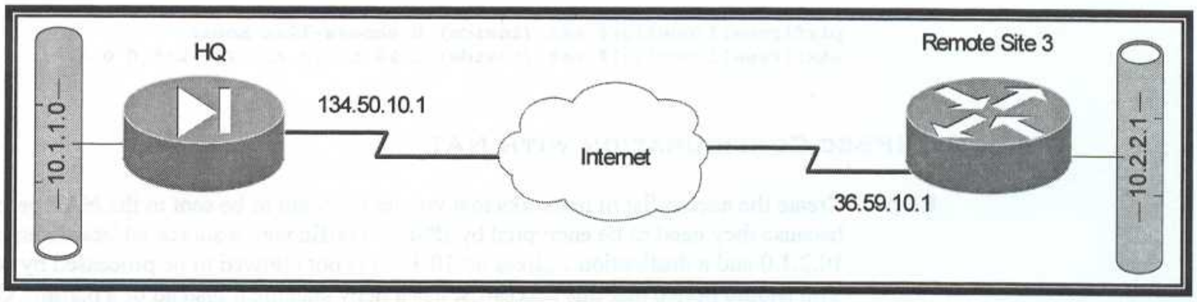
Step 5 Apply NAT to the inside and outside interfaces.

```
hq-vpnrtr(config)# interface FastEthernet0/1
hq-vpn-rtr(config-if)# ip nat inside
hq-vpn-rtr(config)# interface Serial0/0
hq-vpn-rtr(config-if)# ip nat outside
```

Example 5 – PIX to router configuration with DES and Manual keys

In this example, IKE with pre-shared keys is not configured. This is not typical of Cisco environments but may be found when connecting to another vendor's VPN product that does not support IKE. It is also faster since there is no security association negotiation. It is also difficult to configure and prone to typos. Another drawback is the inability to support anti-replay. Anti-replay is a security service where the receiver can reject old or duplicate packets in order to protect itself against replay attacks. IKE provides IPSec with this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS IPSec provides this service whenever it provides the data authentication service.

Figure 18.4 *PIX to router connection*



Headquarters PIX configuration

Step 1 Configure interfaces for outside and inside.

```
hq-pix-vpn(config)# nameif ethernet 0 outside security0
hq-pix-vpn(config)# nameif ethernet1 inside security100
hq-pix-vpn(config)# ip address outside 134.50.10.1 255.255.255.0
hq-pix-vpn(config)# ip address inside 10.1.1.4 255.255.255.0
```

Step 2 Create an access list to define the traffic to protect.

```
hq-pix-vpn(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
10.2.2.0 255.255.255.0
```

Step 3 Configure a transform set that defines how the traffic will be protected. You can configure only one transform set for manually established security associations. The peer must also have the same transform set specified. Our example uses DES for the encryption type and both AH and ESP for the security protocols.

```
hq-pix-vpn(config)# crypto ipsec transform-set weak ah-sha-hmac esp-des
esp-sha-hmac
```

Step 4 Create a crypto map entry in IPSec manual mode.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 ipsec-manual
```

Step 5 Apply the access list to the crypto map. The access list can specify only one permit entry when you are establishing manual security associations.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 match address 100
```

Step 6 Specify the peer to which the IPSec protected traffic can be forwarded. Only one peer can be specified when you are configuring manual security associations.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 set peer 36.59.101.1
```

Step 7 Specify which transform set should be used. This must be the same parameters are specified in the peer's corresponding crypto map entry.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 set transform-set weak
```

Step 8 If the specified transform set includes the AH protocol (hash authentication via MD5-HMAC or SHA-HMAC), set the AH Security Parameter Index (SPI) and key to apply to inbound protected traffic. If you do not plan to use AH skip to step 10. Manual keys must use hex digits.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 set session-key inbound ah
300 728A86CFE210
```

Step 9 Apply the AH SPI's and keys to outbound protected traffic.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 set session-key outbound ah
400 102A39854C34
```

Step 10 If the specified transform set includes the ESP protocol, set the ESP SPI's and keys to apply to inbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
hq-pix-vpn(config)# crypto map mymanualmap 10 set session-key inbound esp
300 cipher  193874A1C143 authenticator  918237FCA1
```

Step 11 Set the ESP SPI's and keys to apply to outbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
hq-pix-vpn(config)# crypto map mymanualmap 30 set session-key outbound esp
400 cipher 10389EA97416 authenticator 293E47520129
```

Step 12 Allow IPSec connections to the PIX.

```
hq-pix-vpn(config)# sysopt connection permit-ipsec
```

Step 13 Apply the crypto map to an interface.

```
hq-pix-vpn(config)# crypto map mymanualmap interface outside
```

REMOTE SITE ROUTER CONFIGURATION

Step 1 Configure the hostname.

```
router(config)# hostname site1-vpn-rtr
```

Step 2 Configure the access-lists.

```
site1-rtr-vpn(config)# access-list 100 permit ip 10.2.2.0 0.0.0.255
10.1.1.0 0.0.0.255
```

Step 3 Configure the transform set.

```
site1-rtr-vpn(config)# crypto ipsec transform-set strong esp-3des esp-sha-
hmac
```

Step 4 Configure the crypto map. Notice that it is now `ipsec-manual` instead of `ipsec-isakmp`.

```
site1-rtr-vpn(config)# crypto map netcg 10 ipsec-manual

% NOTE: This crypto map is incomplete.  Traffic for this or any later
crypto maps will be dropped.
To remedy the situation add a peer and a valid access-list to this crypto
map.
```

Step 5 Configure the peer transform set and access-list.

```
site1-rtr-vpn(config-crypto-map)# set peer 134.50.10.1
site1-rtr-vpn(config-crypto-map)# set transform-set weak
site1-rtr-vpn(config-crypto-map)# match address 100
```

Step 6 Configure the inbound and outbound ESP cipher and authenticator. Configure AH if needed.

```
site1-rtr-vpn(config-crypto-map)# set session-key inbound esp 400 cipher
10389EA97416 authenticator 293E47520129
site1-rtr-vpn(config-crypto-map)# set session-key outbound esp 300 cipher
193874A1C143 authenticator 918237FCA1
site1-rtr-vpn(config-crypto-map)# set session-key inbound ah 400
102A39854C34
site1-rtr-vpn(config-crypto-map)# set session-key outbound ah 300
728A86CFE210
```

Step 7 Configure the interfaces. Apply the crypto map to the outside interface.

```
site1-rtr-vpn(config)#interface FastEthernet0/1
site1-rtr-vpn(config-if)# ip address 10.2.2.1 255.255.255.0
site1-rtr-vpn(config-if)# description Inside Network

site1-rtr-vpn(config)# interface Serial0/0
site1-rtr-vpn(config-if)# description Internet Connection
site1-rtr-vpn(config-if)# ip address 36.59.101.1 255.255.255.252
site1-rtr-vpn(config-if)# crypto map netcg
```

Verification and Troubleshooting

The first step in verifying your VPN is to ping from one network to another. This typically means that you are pinging from one network with unregistered IP addresses to another unregistered network using an Internet VPN. Normally this is not possible since the Internet will block unregistered addresses. However, we now understand that IPSec encapsulates these packets in to a routable packet for transport over the Internet to the IPSec peer. In a lab environment, your entire network may be using unregistered or invalid (on the Internet) addresses. But, the premise of communicating securely from one protected network to another protected network remains the same.

From a router, you can perform an extended ping to verify your VPN. Accept the defaults on all questions except Target IP, Extended commands, and source address to force which interface you want the router to use as its source, otherwise the router will possibly use a different interface and the packet may not be encrypted. If your ping is successful, your VPN is working properly. The PIX does not have extended ping capability.

```
hq-vpn-rtr# ping
Protocol [ip]:
Target IP address: 10.2.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
```

```
Sending 5, 100-byte ICMP Echos to 10.2.1.1, timeout is 2 seconds:
!!!!!
```

If your ping is unsuccessful, there are several steps to take to troubleshoot. Here is a checklist for VPN connectivity. Sample output commands are listed after the checklist.

1. Can you ping the peer VPN router or PIX's outside address? If not, you have Internet connection problems which must be resolved first. There may also be an access-list or firewall filtering pings.
2. Using the **`show crypto ipsec sa`** you can check several potential problems
    a. Check to make sure the crypto map tag is applied to the outside interface.
    b. Check the IP address to verify that the correct IP address is configured on the other VPN device as the peer. It is helpful to have a **`show run`** or **`write terminal`** for both peers to make side-by-side comparisons.
3. Make sure you are pinging from the local network configured in the local ident. In our example below, packets with a source of 192.168.1.0 are being encrypted if the destination is 10.1.1.0. If this does not appear correct check your access-list configured in your crypto map.
4. Check to see if packets are being encrypted or decrypted. If packets are being encrypted only and not decrypted, the problem is likely at the other peer.
5. Verify that the transform-set is the same on both sides. If one side has the wrong encryption type or hash, the VPN will not work.
6. Verify that ISAKMP settings are the same on both sides using the **`show crypto isakmp policy`** command. Also, verify that your ISAKMP key is identical on both peers.
7. If NAT is configured, make sure you block VPN packets from the NAT process.
8. Make sure you have access-lists permitting IPSec traffic or the appropriate **`sysopt connection permit`** commands on the PIX.
9. If you are using RSA with digital certificates verify that your CA is available and working properly.
10. Rebooting the router or PIX after configuring an IPSec VPN is a good idea as long as the configuration is saved to NVRAM and the device is not yet passing production traffic.

```
hq-vpn-rtr# show crypto ipsec sa
interface: FastEthernet0/0
Crypto map tag: netcg, local addr. 64.107.35.1

local  ident (addr/mask/port/port): (192.168.1.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.1.1.0/255.255.255.0/0/0)
current_peer: 134.50.10.1
PERMIT, flags={origin_is_acl,}
#pkts encaps: 1276, #pkts encrypt: 1276, #pkts digest 1276
#pkts decaps: 1165, #pkts decrypt: 1165, #pkts verify 1165
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 3, #recv errors 0

local crypto endpt.: 64.107.35.1, remote crypto endpt.: 134.50.10.1
path mtu 1500, media mtu 1500
current outbound spi: D47684DF

inbound esp sas:
spi: 0x82AE809(137029641)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2070, flow_id: 71, crypto map: netcg
sa timing: remaining key lifetime (k/sec): (4607997/24956)
IV size: 8 bytes
replay detection support: Y

inbound ah sas:

inbound pcp sas:
```

```
outbound esp sas:
spi: 0xD47684DF(3564537055)
transform: esp-3des esp-sha-hmac ,
in use settings ={Tunnel, }
slot: 0, conn id: 2071, flow_id: 72, crypto map: netcg
sa timing: remaining key lifetime (k/sec): (4607998/24956)
IV size: 8 bytes
replay detection support: Y

outbound ah sas:

outbound pcp sas:

hq-vpn-rtr# show crypto isakmp policy
Protection suite of priority 10
        encryption algorithm:   Three key triple DES
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit

Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit

hq-vpn-rtr# show crypto isakmp key
Hostname/Address        Preshared Key
134.50.10.1           ciscovpnkey1
```

Typical Gotchas!

- Trying to apply two different crypto map names to a single interface
- Mismatched ACLs for encrypted traffic on the two VPN devices
- VPN traffic getting NAT'd
- Mismatched ISAKMP values or keys
- Mismatched IPSec values
- VPN device blocking VPN ports (IP protocol 50, 51, or UDP 500)
- No IP routing reachability to VPN networks

CHAPTER 19

CERTIFICATE AUTHORITY

MICROSOFT CA CONFIGURATION

Install Windows 2000 Server and Service Pack 1. Do not install IIS or Certificate Authority service until SP1 is installed. If IIS and CA are installed without SP1, uninstall them and reinstall after applying SP1.

SETUP AND INSTALL CERTIFICATE AUTHORITY ON WINDOWS 2000

Step 1 Logon as Administrator or a user that has administrator privileges.

Step 2 Click *Start>Settings>ControlPanel>Add/Remove Programs>Add/Remove Windows Components.* This is where you'll find the option to add both IIS and Certificate Services.

Step 3 Install Internet Information Server (IIS), reboot when done. Accepting the defaults during the installation should be sufficient. For more information on IIS, see Microsoft's website.

Step 4 Install Certificate Services. Once you have installed CA, you cannot change the name of the server, join a domain or be removed from a domain.

Step 5 Select Standalone CA and fill out the information it requests. Allow IIS to be stopped when prompted.

Step 6 Reboot the server.

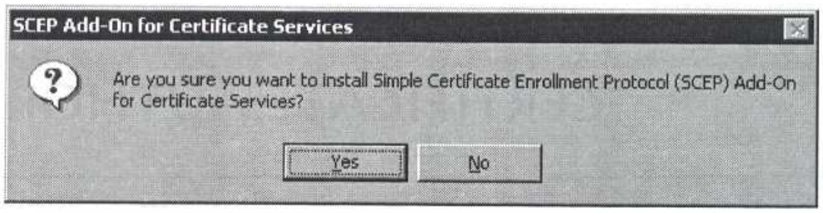
Step 7 Open a browser and point to the CA server URL such as http://NLI-SECURITY1/certsrv/ to verify your CA is running. If you do not have DNS running you may want to use the IP address instead of the name.

INSTALL SIMPLE CERTIFICATE ENROLLMENT PROTOCOL

In order for Cisco routers and PIX firewalls to use the CA server, you must install the SCEP package. This package is found on the Microsoft Resource Kit.

Step 1 Download or locate the file scep.exe. Launch the installation. Click *Yes* to install.

Figure 19.1 *SCEP Install*

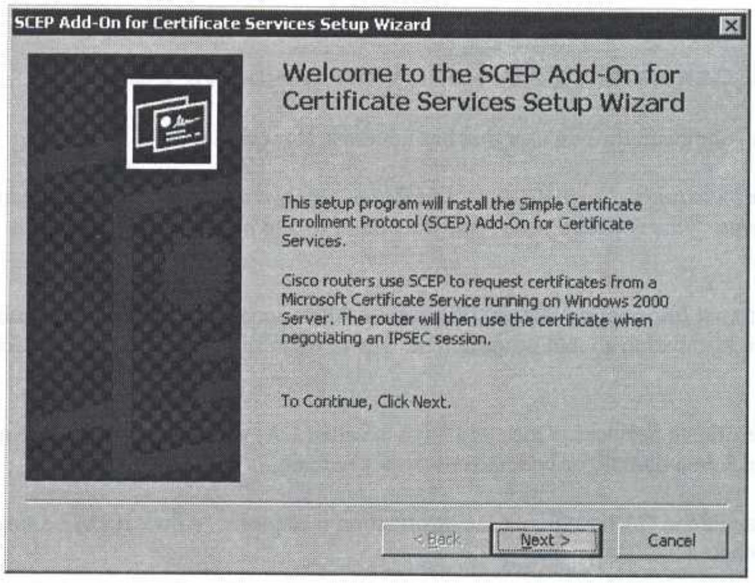


SCEP Add-On for Certificate Services

Are you sure you want to install Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services?

Step 2 Click *Next* to continue.

Figure 19.2 *SCEP Setup Wizard*



SCEP Add-On for Certificate Services Setup Wizard

Welcome to the SCEP Add-On for Certificate Services Setup Wizard

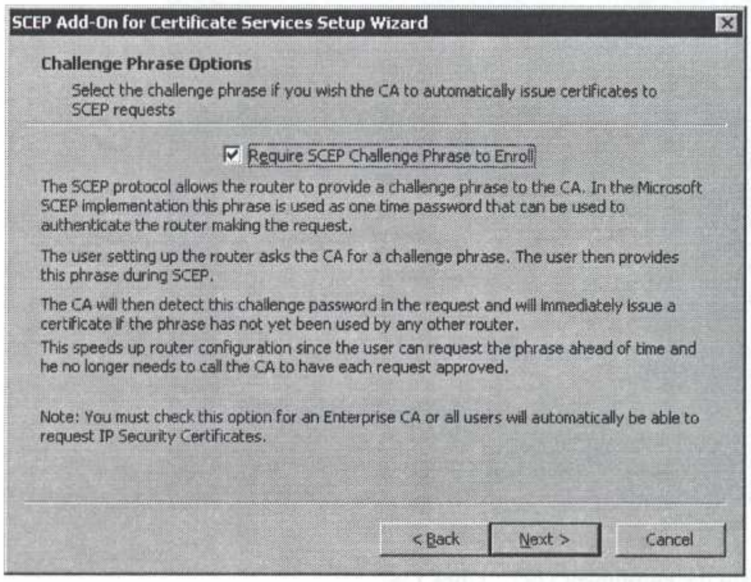
This setup program will install the Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services.

Cisco routers use SCEP to request certificates from a Microsoft Certificate Service running on Windows 2000 Server. The router will then use the certificate when negotiating an IPSEC session.

To Continue, Click Next.

Step 3 This option should be left checked.

Figure 19.3 *SCEP Challenge Phrase Options*



SCEP Add-On for Certificate Services Setup Wizard

Challenge Phrase Options

Select the challenge phrase if you wish the CA to automatically issue certificates to SCEP requests

☑ Require SCEP Challenge Phrase to Enroll

The SCEP protocol allows the router to provide a challenge phrase to the CA. In the Microsoft SCEP implementation this phrase is used as one time password that can be used to authenticate the router making the request.

The user setting up the router asks the CA for a challenge phrase. The user then provides this phrase during SCEP.

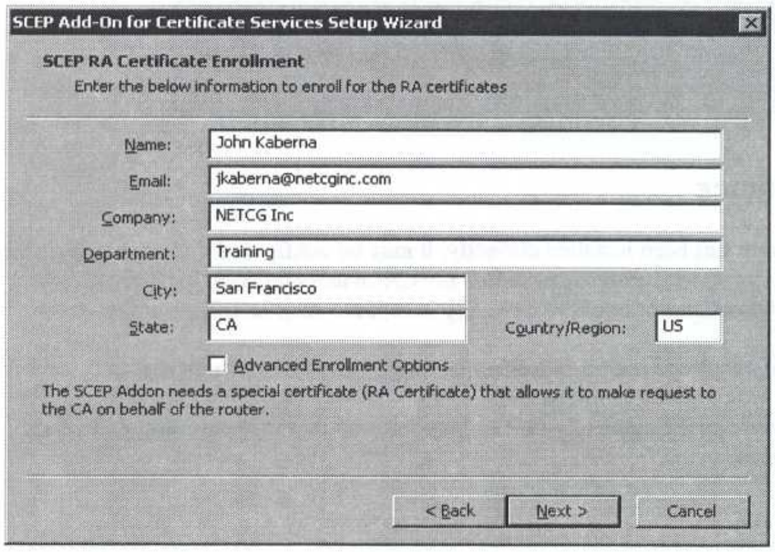
The CA will then detect this challenge password in the request and will immediately issue a certificate if the phrase has not yet been used by any other router.

This speeds up router configuration since the user can request the phrase ahead of time and he no longer needs to call the CA to have each request approved.

Note: You must check this option for an Enterprise CA or all users will automatically be able to request IP Security Certificates.

Step 4 Enter the information requested.

Figure 19.4 *SCEP RA Certificate Enrollment*



SCEP Add-On for Certificate Services Setup Wizard

SCEP RA Certificate Enrollment

Enter the below information to enroll for the RA certificates

Name: John Kaberna

Email: jkaberna@netcginc.com

Company: NETCG Inc

Department: Training

City: San Francisco

State: CA

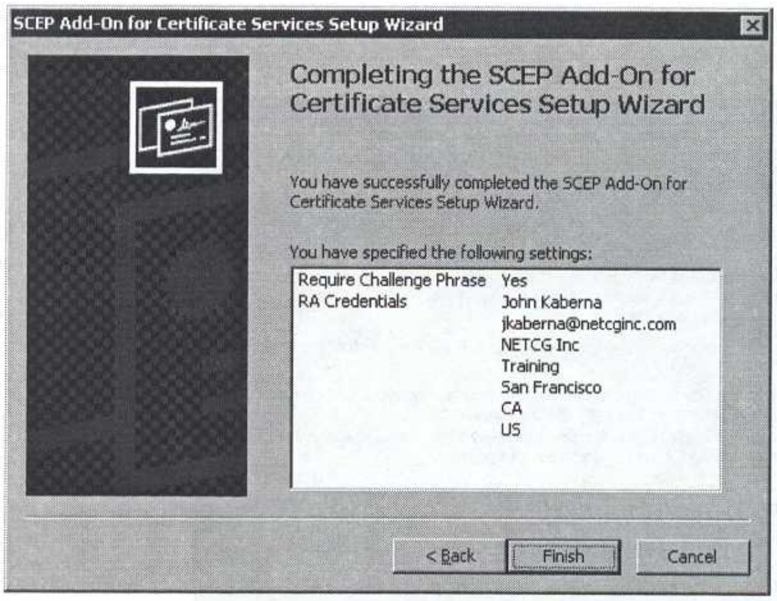
Country/Region: US

□ Advanced Enrollment Options

The SCEP Addon needs a special certificate (RA Certificate) that allows it to make request to the CA on behalf of the router.

Step 5 Click *Finish.*

Figure 19.5 *Completing the SCEP Setup Wizard*



Completing the SCEP Add-On for Certificate Services Setup Wizard

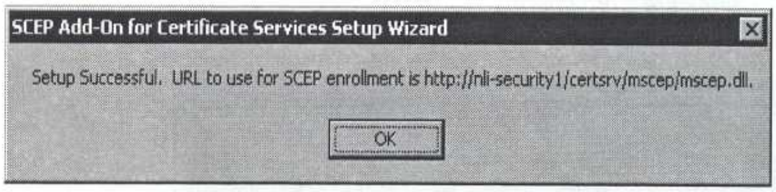
You have successfully completed the SCEP Add-On for Certificate Services Setup Wizard.

You have specified the following settings:

```
Require Challenge Phrase  Yes
RA Credentials            John Kaberna
                          jkaberna@netcginc.com
                          NETCG Inc
                          Training
                          San Francisco
                          CA
                          US
```

Step 6 Write down the enrollment URL. Click *OK.*

Figure 19.6 *SCEP Setup Successful and Enrollment URL*



Setup Successful. URL to use for SCEP enrollment is http://nli-security1/certsrv/mscep/mscep.dll.

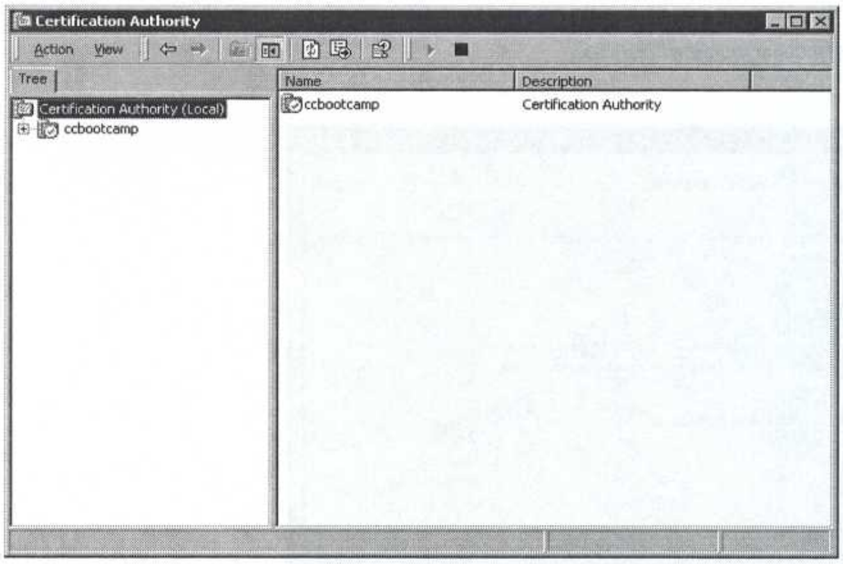
SETUP THE CA SERVICE

Now that the server has been installed correctly, it may be configured if changes are necessary. One of the most common changes is to switch the CA to automatically issue certificates. Otherwise, the Administrator needs to manually accept the each certificate at the server.

Step 1 Click *Start>Programs>Administrative Tools>Certificate Authority.*

Step 2 Right click on the name of your CA (ccbootcamp in our screen shot) and select *Properties.*

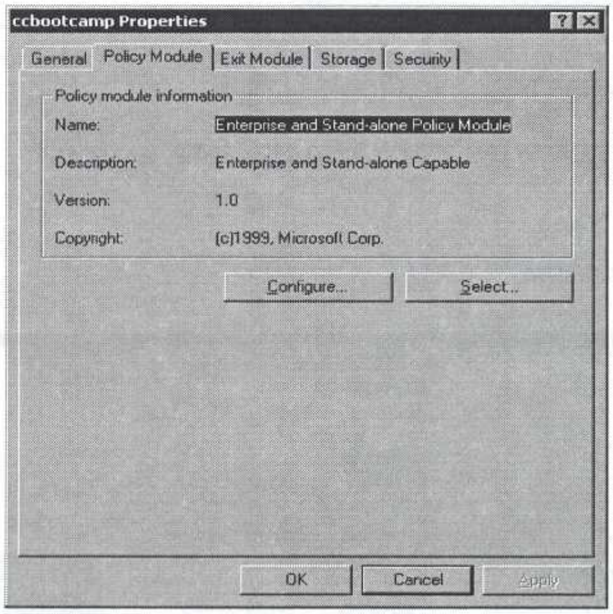
Figure 19.7 *Certificate Authority main window*



Name	Description
ccbootcamp	Certification Authority

Step 3 Click the *Policy Module* tab and then the *Configure* button.

Figure 19.8 *Policy Module tab*



Name: Enterprise and Stand-alone Policy Module

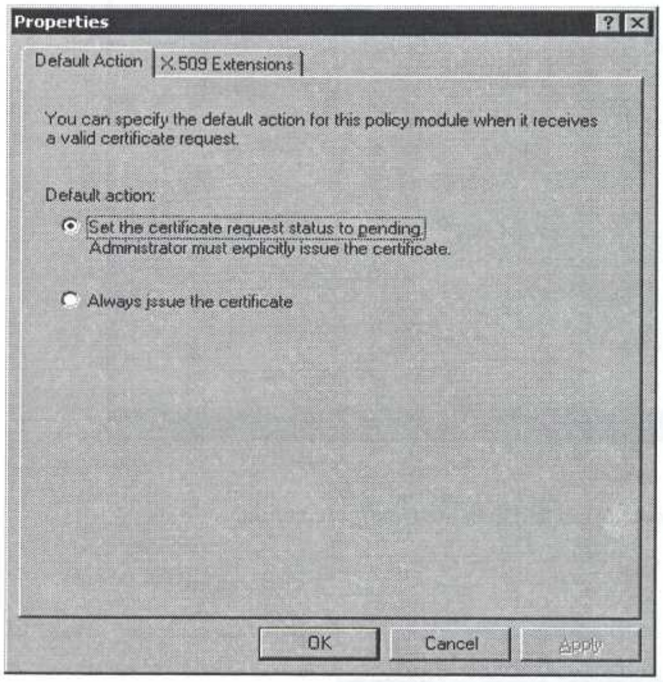
Description: Enterprise and Stand-alone Capable

Version: 1.0

Copyright: (c)1999, Microsoft Corp.

Step 4 Click the radio button next to *Always issue the certificate.* If your CA server has trouble issuing certificates, you can change this option back to pending.

Figure 19.9. *Certificate issue action tab*



Properties

Default Action | X.509 Extensions

You can specify the default action for this policy module when it receives a valid certificate request.

Default action:

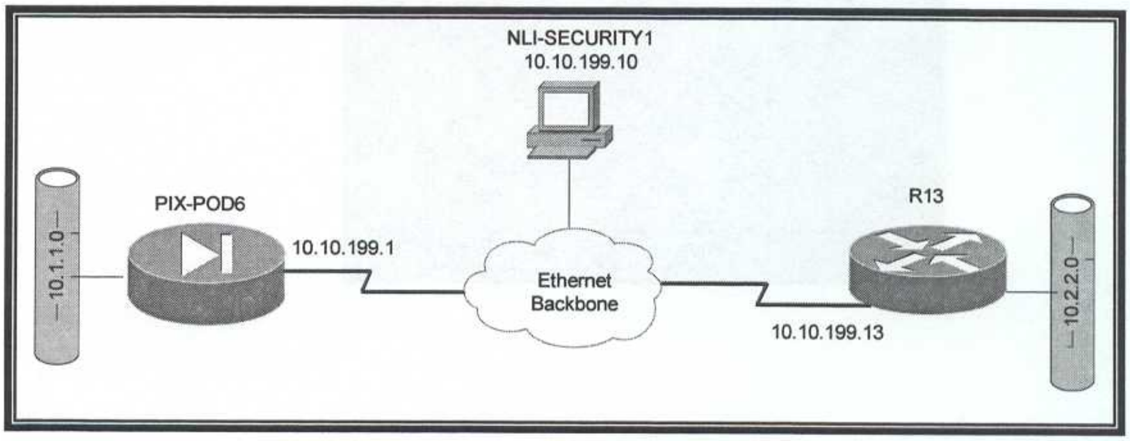
● Set the certificate request status to pending. Administrator must explicitly issue the certificate.

○ Always issue the certificate

Router and PIX configuration

Example 1 – Router to PIX VPN with 3DES and IKE RSA digital certificates

Figure 19.10 *Router to PIX VPN connection*



HEADQUARTERS ROUTER CONFIGURATION

Step 1 Configure ISAKMP policies. The only difference between a pre-shared key ISAKMP policy and RSA is the authentication type.

```
r13(config)# crypto isakmp policy 10
r13(config-isakmp)# encryption 3des
r13(config-isakmp)# authentication rsa-sig
r13(config-isakmp)# hash sha
r13(config-isakmp)# lifetime 86400
```

Step 2 Configure the router for the proper time. If you are using NTP, make sure your server is also using NTP. It is recommended that you use GMT time.

```
r13# clock set 01:31:00 22 March 2002
r13(config)# clock timezone GMT -0
```

Step 3 The next 4 steps configure the Certificate Authority Server parameters. In this case, our server is nli-security1 and the IP address is 10.10.199.10. The `ip host` command is a basic static host to IP address translation command if you do not have DNS configured or if the server's IP address is not listed in DNS.

```
r13(config)# ip domain-name ccbootcamp.com
r13(config)# ip host nli-security1 10.10.199.10
r13(config)# ip name-server 10.10.199.10
```

Step 4 Create your `crypto ca identity` using an arbitrary name. It is recommended that you use your domain name.

```
r13(config)# crypto ca identity ccbootcamp
```

Step 5 Configure your enrollment URL and mode. RA stands for Registration Authority and is required for the Microsoft CA server.

```
r13(ca-identity)# enrollment url
http://nli-security1:80/certsrv/mscep/mscep.dll
r13(ca-identity)# enrollment mode ra
```

Step 6 Configure the router to not require the Certificate Revocation List (CRL) from the CA. The CRL is a list of certificates that have expired and are no longer valid. However, if for some reason the CRL is not available or the CA is down, the router will reject ALL certificates. The `crl optional` command is required when using a Microsoft CA server.

```
r13(ca-identity)# crl optional
```

Step 7 Retrieve the Certificate of the CA by entering the following command and entering yes when prompted to a accept the certificate.

```
r13(config)# crypto ca authenticate ccbootcamp
Certificate has the following attributes:
C192837B EF9263A5  819237FE 19A2B5F0

% Do you accept this certificate? [yes/no]: yes
```

Step 8 Generate public and private keys by using the following `crypto key` command and then selecting the modulus when prompted. Depending on your IOS version and feature set, you may also get a message that SSH is enabled.

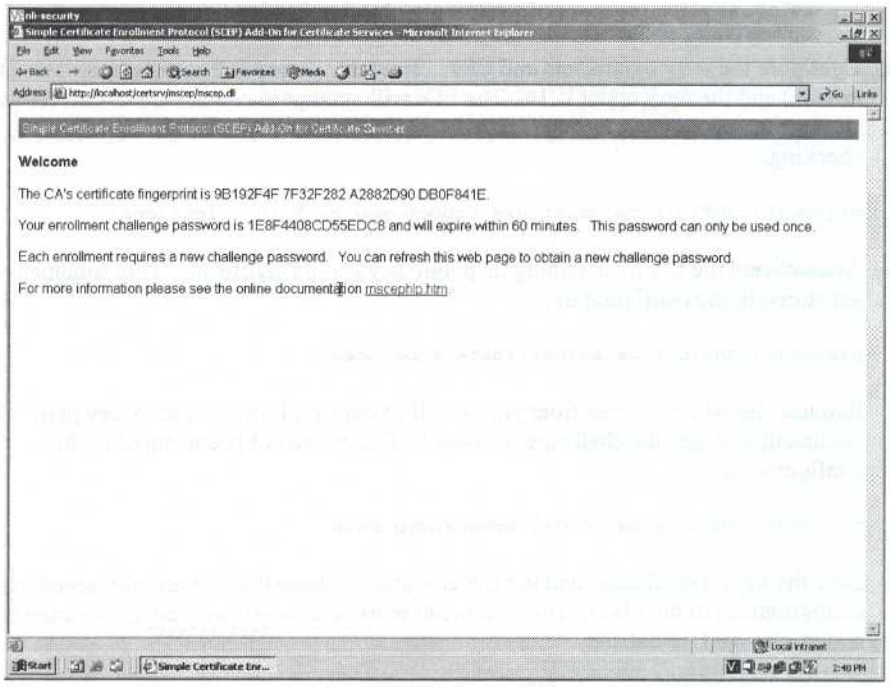
```
r13(config)# crypto key generate rsa usage-keys
The name for the keys will be: ca-router.ccbootcamp.com
```

```
Choose the size of the key modulus in the range of 360 to 2048 for your
  Signature Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 512
Generating RSA keys ...
[OK]
Choose the size of the key modulus in the range of 360 to 2048 for your
  Encryption Keys. Choosing a key modulus greater than 512 may take
  a few minutes.

How many bits in the modulus [512]: 512
Generating RSA keys ...
[OK]

r13(config)#
00:06:06: %SSH-5-ENABLED: SSH 1.5 has been enabled
ca-router(config)#
```

Step 9 Submit the router's public key(s) to the CA server and request a certificate. You will be prompted for several questions. Check with your CA provider (the proctor in the CCIE lab) for any requirements such as router serial number or IP address. At minimum, you will need to enter the password and choose yes to request the certificate. The sample below will work with a Microsoft CA server. Depending on your settings for issuing certificates (pending or always issue), you may need to enter the password provided by the CA server.

```
r13(config)# crypto ca enroll ccbootcamp
% Start certificate enrollment ..
% Create a challenge password.  You will need to verbally provide this
password to the CA administrator in order to revoke your certificate.  For
security reasons your password will not be saved in the configuration.
Please make a note of it.

Password: 1E8F4408CD55EDC8
Re-enter password: 1E8F4408CD55EDC8
% The subject name in the certificate will be: r13.ccbootcamp.com
% Include the router serial number in the subject name ? [yes/no]: no
% Include an IP address in the subject name? [yes/no]: no
Request certificate from CA? [yes/no]: yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show cyrpto ca certificate' command will also show the fingerprint.

Signing Certificate Request Fingerprint:
127834AB 1938BF19 38EC9816 AA001237
Encryption Certificate Request Fingerprint:
19235FA9 A3C09816 A0A01337 78EC34AB
```

Figure 19.11 *CA Server Password Page*



nli-security
Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost/certsrv/mscep/mscep.dll
Go Links
Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services
Local intranet
Start
Simple Certificate Enr...
2:40 PM

Welcome

The CA's certificate fingerprint is 9B192F4F 7F32F282 A2882D90 DB0F841E.

Your enrollment challenge password is 1E8F4408CD55EDC8 and will expire within 60 minutes. This password can only be used once.

Each enrollment requires a new challenge password. You can refresh this web page to obtain a new challenge password.

For more information please see the online documentation mscephlp.htm.

Step 10 Configure the crypto map policies, access-list, and transform set.

```
r13(config)# access-list 100 permit ip 10.2.2.0 0.0.0.255 10.1.1.0
    0.0.0.255
r13(config)# crypto ipsec transform-set weak esp-des esp-sha-hmac
r13(config)# crypto map toPIX 10 ipsec-isakmp
r13(config-crypto-map)# set peer 10.10.199.1
r13(config-crypto-map)# set transform-set weak
r13(config-crypto-map)# match address 100
```

Remote PIX configuration

This example is for a Microsoft based certificate server. This type of server is the easiest to setup and does not require a 3rd party vendor. It is also the most likely server you will see on the lab exam. Each vendor will have specific requirements so if you are implementing another type check the latest documentation.

Step 1 Configure the PIX host name. The hostname should be unique for the domain.

```
pixfirewall(config)# hostname pix-pod6
```

Step 2 Configure the PIX domain name.

```
pix-pod6(config)# domain-name ccbootcamp.com
```

Step 3 Generate the PIX RSA key pair(s). The 512 represents the modulus, or size of the key.

```
pix-pod6(config)# ca generate rsa key 512
```

Step 4 Define the identity parameters. The PIX must be instructed where to find the certificate.

```
pix-pod6(config)# ca identity ccbootcamp
    10.10.199.10:/certsrv/mscep/mscep.dll
```

Step 5 Configure the retry parameters and CRL. The retry period is 2 minutes (the default is one minute) and the retry count is 10. The PIX will attempt to contact the CA server every 2 minutes for 10 attempts before it fails. The `crloptional` option disables the CRL checking.

```
pix-pod6(config)# ca configure ccbootcamp ra 2 10 crloptional
```

Step 6 Authenticate the CA by obtaining its public key and its certificate. This command does not get stored in the configuration.

```
pix-pod6(config)# ca authenticate ccbootcamp
```

Step 7 Request signed certificates from your CA for your PIX Firewall's RSA key pair. This command also sets the challenge password. This password is not stored in the configuration

```
pix-pod6(config)# ca enroll ccbootcamp ccie
```

Step 8 Save the keys, certificates, and the CA commands (those that are actually saved in the configuration) to the Flash. This command must be entered any time ca commands are added, changed, or deleted.

```
pix-pod6(config)# ca save all
```

Step 9 Configure an IKE policy and apply it to the outside interface.

```
pix-pod6(config)# isakmp policy 10 auth rsa-sig
pix-pod6(config)# isakmp enable outside
```

Step 10 Configure the access-list of traffic to protect using IPSec.

```
pix-pod6(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
    10.2.2.0 255.255.255.0
```

Step 11 Configure a transform set.

```
pix-pod6(config)# crypto ipsec transform-set weak esp-des esp-sha-hmac
```

Step 12 Define a crypto map.

```
pix-pod6(config)# crypto map mymap 10 ipsec-isakmp
pix-pod6(config)# crypto map mymap 10 match address 100
pix-pod6(config)# crypto map mymap 10 set transform-set weak
pix-pod6(config)# crypto map mymap 10 set peer 10.10.199.13
```

Step 13 Apply the crypto map to the outside interface.

```
pix-pod6(config)# crypto map mymap interface outside
```

Step 14 Configure the PIX Firewall to implicitly permit IPSec traffi

```
pix-pod6(config)# sysopt connection permit-ipsec
```

Typical Gotchas!

- Installation appears successful, but a reinstall fixes unexplainable problems
- Clock not set on VPN devices or clock is set earlier than the CA server
- Entering an arbitrary password instead of the one received via a web browser to the mscep.dll on the CA server

CHAPTER 20

POINT-TO-POINT TUNNELING PROTOCOL

Point-to-Point Tunneling Protocol (PPTP) is a layer 2 tunneling protocol that allows clients to communicate securely over public networks such as the Internet. PPTP is described in RFC 2637. Cisco began support of PPTP with IOS release 12.1(5)T for most platforms. The Cisco PIX also supports PPTP beginning with software release 5.1. PPTP is configured under Virtual Private Dialup Network (VPDN) commands. Other protocols such as Layer 2 Tunneling Protocol (L2TP) and Layer 2 Forwarding (L2F) also use VPDN commands. These protocols are discussed in the next chapter.

VPDN users can be authenticated using either local, TACACS+, or RADIUS authentication. In order to test our configurations, configure a Windows PC for PPTP. The next section gives a basic description on how to configure a VPN on the different Windows PC's. This section assumes that LAN Internet access or Dial-Up Networking is configured. For additional information and troubleshooting on Windows PC's check Microsoft's website.

Windows 98/NT/2000 Configurations for PPTP

Windows 98 Configuration

Step 1 Install the PPTP feature.

1) Select *Start > Settings > Control Panel > Add New Hardware*. Click *Next*.
2) Click *Select from List* and choose *Network Adapter*. Click *Next*.
3) Choose *Microsoft* in the left panel and *Microsoft VPN Adapter* on the right panel.

Step 2 Configure the PPTP feature.

1) Select *Start > Programs > Accessories > Communications > Dial Up Networking.*
2) *Click Make new connection* and for *Select a device*, connect using Microsoft VPN Adapter. The VPN Server IP address is the PIX tunnel endpoint.
3) The Windows 98 default authentication is to use password encryption, that is, CHAP or MS-CHAP. To change the PC to also allow PAP: select *Properties > Server types*. Uncheck *Require encrypted password.* You can configure data encryption (MPPE) in this area.

Windows 2000 Configuration

Step 1 *Select Start > Programs > Accessories > Communications > Network & Dialup connections.*

Step 2 Click *Make new connection* and then click Next.

Step 3 Select *Connect to a private network through the Internet* and Dial a connection prior (or not if LAN). Click *Next.*

Step 4 Enter the hostname or IP address of tunnel endpoint (PIX/router).

Step 5 If there's a need to change the password type, select *Properties > Security for the connection > Advanced.* The default is MS-CHAP and MS-CHAP v2 (not CHAP or PAP). You can configure data encryption (MPPE or no MPPE) in this area.

Windows NT 4.0 Configuration

Step 1 Configure the PPTP Protocol

1) Click *Start > Settings > Control Panel > Network > Protocols.*
2) In the Protocols box, click *Add to display the Select Network Protocol* dialog box. Select *Point To Point Tunneling Protocol* and click *OK.*
3) Type the drive and directory location of your installation files in the Windows NT Setup dialog box, and then click Continue. The PPTP files are copied from the installation directory and the PPTP Configuration dialog box will appear.
4) Click the Number of Virtual Private Networks drop-down arrow and select the number of VPN devices you want the client to support. You can select a number between 1 and 256 for computers running Windows NT Workstation version 4.0 or Windows NT Server version 4.0. Typically, only one VPN device is installed on a PPTP client.
5) Click *OK,* and then click *OK* in the Setup Message dialog box.
6) In the Remote Access Setup properties dialog box continue installation by clicking *Add* to add to RAS the VPN device installed with PPTP.

Step 2 Add a VPN Device as a RAS Port on the PPTP Client. You must add the VPN device to RAS after installing PPTP. Follow these steps to add a VPN device on a computer running Windows NT Workstation version 4.0.

1) Click *Start > Settings > Control Panel.*
2) In Control Panel, double-click *Network.*
3) Click the *Services* tab and select *Remote Access Service.*
4) Click *Properties* to display the Remote Access Setup properties dialog box.
5) Click *Add.*
6) Click the *RAS Capable Devices* list to display the VPN devices that must be added and configured as a port and device in RAS.
7) Select the *VPN1 - RASPPTPM* device, and then click *OK.* (If you installed PPTP with more than one VPN device, repeat steps 5, 6, and 7 until all the VPNs are added to the Remote Access Setup properties dialog box.)
8) By default, the VPN device on a computer running Windows NT Workstation version 4.0 is configured to dial out only. Select the VPN port and click *Configure.* Verify that the Dial out only option in the Port Usage dialog box is the only option selected, and then click *OK.* This returns you to the Remote Access Setup properties dialog box.
9) Click *Network* to display the Network Configuration dialog box.
10) Verify that the TCP/IP option in Dial out Protocols is the only option checked, and then click *OK.*

11) Click *Continue.*
12) Close *Network,* and then restart the computer.

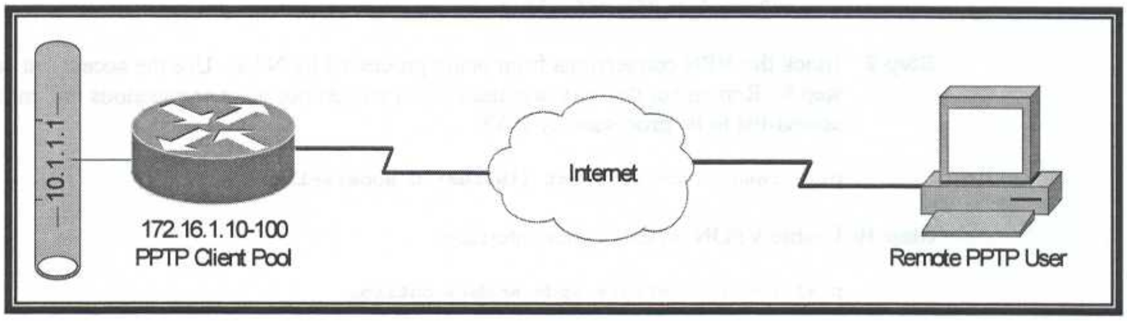
PIX Configuration for PPTP

The following four examples give the commands needed to configure most PPTP services. After completing this section, the candidate should be able to perform additional tasks such as use TACACS+ instead of RADIUS for authentication or configure a router for local authentication and no encryption.

Example 1 – PIX Configuration with Local Authentication and no encryption
Example 2 – PIX Configuration with Local Authentication and encryption
Example 3 – PIX Configuration with RADIUS Authentication and encryption
Example 4 – Router configuration with RADIUS Authentication and encryption

Example 1 – PIX Configuration with Local Authentication and No Encryption

Figure 20.1 *PIX, PPTP, and local authentication*



Step 1 Configure the address pool for incoming clients. All incoming VPDN connections will be dynamically assigned an IP address from the pool. The PIX internal network uses 10.1.1.0. Even though the PIX does not have an interface in the 172.16.1.0 network it can still route this traffic.

```
pixfirewall(config)# ip local pool pptp-pool 172.16.1.10-172.16.1.100
```

Step 2 Allow PPTP connections through the firewall using the **sysopt connection permit-pptp** command.

```
pixfirewall(config)# sysopt connection permit-pptp
```

Step 3 Configure the PIX to accept dial-in PPTP connections.

```
pixfirewall(config)# vpdn group1 ppp accept dialin pptp
```

Step 4 Configure the authentication methods (CHAP, MSCHAP, or PAP). The PIX can be configured for any combination of methods or even all three simultaneously as shown below.

```
pixfirewall(config)# vpdn group 1 ppp authentication chap
pixfirewall(config)# vpdn group 1 ppp authentication mschap
pixfirewall(config)# vpdn group 1 ppp authentication pap
```

Step 5 Configure VPDN to use the address pool created in Step 1.

```
pixfirewall(config)# vpdn group 1 client configuration address local pptp-
pool
```

Step 6 Configure VPDN to use local authentication.

```
pixfirewall(config)# vpdn group 1 client authentication local
```

Step 7 Configure local user accounts. This example creates an account for `jdoe` with a password of `ccie`.

```
pixfirewall(config)# vpdn username jdoe password ccie
```

Step 8 Configure an access-list to be used for the **`nat 0`** command. All inbound VPN connection will be terminated properly. However, the return traffic from the internal network 10.1.1.0 will be processed by NAT without Steps 8 and 9. Remember that routers and PIX firewalls process NAT prior to IPSec. By blocking these packets from the NAT process, they will then be handled by the VPDN process.

```
pixfirewall(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
    172.16.1.0 255.255.255.0
```

Step 9 Block the VPN connections from being processed by NAT. Use the access-list created in step 8. Remember that **`nat 0`** is used when you do not want connections that match the access-list to be processed by NAT.

```
pixfirewall(config)# nat (inside) 0 access-list 100
```

Step 10 Enable VPDN on the outside interface.

```
pixfirewall(config)# vpdn enable outside
```

Example 2 - PIX Configuration with Local Authentication and Encryption

This configuration will be the same as the previous example adding Microsoft Point-to-Point Encryption (MPPE). MPPE is capable of operating at either 40-bit or 128-bit encryption. The 128-bit encryption will require the 3DES license to be installed on the PIX. To determine if your PIX is capable of 3DES use the **`show version`** command.

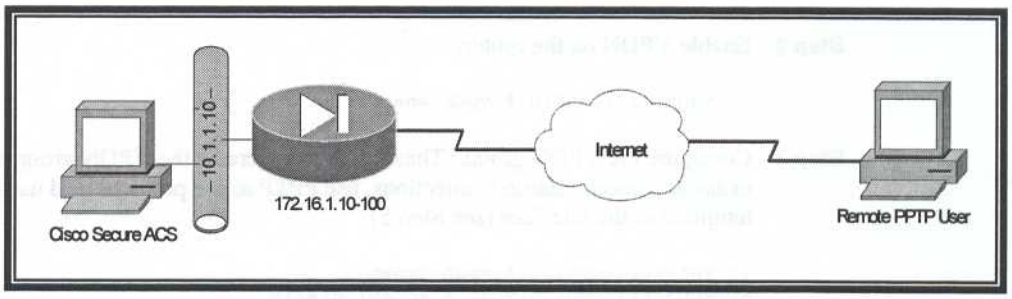
The following command sets the PIX to auto detect MPPE encryption type (40 or 128-bit) and require MPPE. The `require` argument is optional but may be required in your environment based on your security policy. With the `require` option set, if the VPDN Client PC is not configured for MPPE the authentication will fail.

```
pixfirewall(config)# vpdn group 1 ppp encryption mppe auto require
```

Example 3 - PIX Configuration with RADIUS Authentication and Encryption

This configuration requires all the same commands as Example 1 except RADIUS is used for authentication instead of local authentication.

Figure 20.2 *PPTP and RADIUS Authentication*



Step 1 Configure the basic VPDN commands as seen below. Note that we changed the MPPE settings to only work with 40-bit encryption and do not require MPPE be used at all. If MPPE is used it will only work with 40-bit level encryption.

```
pixfirewall(config)# ip local pool pptp-pool 172.16.1.10-172.16.1.100
pixfirewall(config)# sysopt connection permit-pptp
pixfirewall(config)# vpdn group1 ppp accept dialin pptp
pixfirewall(config)# vpdn group 1 ppp authentication chap
pixfirewall(config)# vpdn group 1 ppp authentication mschap
pixfirewall(config)# vpdn group 1 ppp authentication pap
pixfirewall(config)# vpdn group 1 client configuration address local pptp-
    pool
pixfirewall(config)# access-list 100 permit ip 10.1.1.0 255.255.255.0
    172.16.1.0 255.255.255.0
pixfirewall(config)# nat (inside) 0 access-list 100
pixfirewall(config)# vpdn group 1 ppp encryption mppe 40
```

Step 2 Configure basic AAA. The word "pptpauth" is an arbitrary name and can be any word to describe this particular connection. In this example, pptpauth is using the protocol RADIUS. The second command gives the RADIUS host IP address (which is an inside address), password, and timeout value. The same commands can be used to configure TACACS+ by changing the protocol.

```
pixfirewall(config)# aaa-server pptpauth protocol radius
pixfirewall(config)# aaa-server pptpauth (inside) host 10.1.1.10 cisco
    timeout 5
```

Step 3 Configure the VPDN to use RADIUS authentication.

```
pixfirewall(config)# vpdn group 1 client authentication aaa pptpauth
```

Step 4 Enable VPDN on the outside interface.

```
pixfirewall(config)# vpdn enable outside
```

Example 4 - Router Configuration with RADIUS Authentication and Encryption

These commands are significantly different in syntax from the PIX commands, however, the basic concept remains the same. Figure 19-2 is still used for this example, except instead of a PIX Firewall we are using a router.

Step 1 Configure basic AAA commands to use RADIUS and optionally local authentication or any other alternate authentication methods.

```
sf-vpn-rtr1(config)# aaa authentication ppp default group radius local
```

Step 2 Enable VPDN on the router.

```
sf-vpn-rtr1(config)# vpdn enable
```

Step 3 Configure the VPDN group. These commands create the VPDN group name ("1" in this example), accept dial-in connections, use PPTP as the protocol, and use the virtual-template as the interface (see Step 5)

```
sf-vpn-rtr1(config)# vpdn-group 1
sf-vpn-rtr1(config-vpdn)# accept-dialin
sf-vpn-rtr1(config-vpdn-acc-i)# protocol pptp
sf-vpn-rtr1(config-vpdn-acc-i)# virtual-template 1
```

Step 4 Configure a local pool of IP address to be dynamically allocated to incoming VPDN connections.

```
sf-vpn-rtr1(config)# ip local pool pptppool 10.1.1.200 10.1.1.250
```

Step 5 Configure the Virtual-template interface to be used for VPDN. Configure this interface to use the internal IP address using the **ip unnumbered** command. There is the option to use either DHCP or a local pool for incoming connections. In our case, we decided to use a local pool.

```
sf-vpn-rtr1(config)# interface virtual-template 1
sf-vpn-rtr1(config-if)# ip unnumbered fa0/1
sf-vpn-rtr1(config-if)# peer default ip address ?
dhcp  Use DHCP proxy client mechanism to allocate a peer IP address
pool  Use IP pool mechanism to allocate a peer IP address
sf-vpn-rtr1(config-if)# peer default ip add pool pptppool
```

Step 6 Configure authentication types accepted. There are several arguments to the **ppp authentication** command.

```
sf-vpn-rtr1(config-if)# ppp authentication chap ?
callback  Authenticate remote on callback only
callin    Authenticate remote on incoming call only
callout   Authenticate remote on outgoing call only
default   Use the default authentication list
ms-chap   Microsoft Challenge Handshake Authentication Protocol (MS-CHAP)
one-time  Allow use of username*OTP for one-time passwords
optional  Allow peer to refuse to authenticate
pap         Password Authentication Protocol (PAP)

sf-vpn-rtr1(config-if)# ppp authentication chap ms-chap pap
```

Step 7 Configure the RADIUS host and key.

```
sf-vpn-rtr1(config)# radius-server host 192.168.0.10
sf-vpn-rtr1(config)# radius-server host 192.168.0.10 key ccie
```

Troubleshooting PPTP Connections

1. Verify general connectivity. Your PC should be able to reach the outside address of the VPDN access device such as a PIX or router. A ping should work unless ICMP is blocked by a firewall or access-list.
2. Make sure that there are no firewalls or access-lists blocking GRE (port 47) and PPTP (TCP port 1723).
3. Check the bug list to make sure there are no compatibility issues with the IOS or PIX software and the authentication server being used. For example, some IOS versions have trouble with MPPE and CiscoSecure ACS 2.5 for NT.
4. Make sure your platforms actually support the configuration planned. For example, MPPE is not supported in CiscoSecure ACS for UNIX as of this writing. In addition, MS-CHAP version 2 is not supported by VPDN on Cisco devices. Check Cisco's website for enhancements and new features that enable previously unsupported technology.
5. If you are configuring RADIUS or TACACS+ and are not certain whether or not they are working properly, configure the VPDN access point for local authentication. If local authentication works properly then the issue is likely with the AAA configuration on either the NAS or the AAA server itself.
6. Disable encryption (MPPE) initially on both the PC and the VPDN access point. Remember that some versions of CiscoSecure and IOS or PIX software may not support MPPE properly.
7. Windows 98 and later use the username configured in Dial-Up Networking (DUN) for the VPDN connection. If your DUN username and VPDN username are different authentication will not work.
8. With some versions of Windows using MS-CHAP, the PC appends the NT domain name to the front of the username. For example, a username of *jdoe* in NT domain *CISCO* will send *CISCO\jdoe* as the username. If the VPDN access device is not configured with a username of *CISCO\jdoe* authentication will fail. MS-CHAP users will need to have this style of username configured on the VPDN access device or AAA server (if configured).

Typical Gotchas!

- Mismatched encryption type or level on client and VPN device
- AAA not configured properly when using RADIUS or TACACS servers
- Including domain name during user authentication

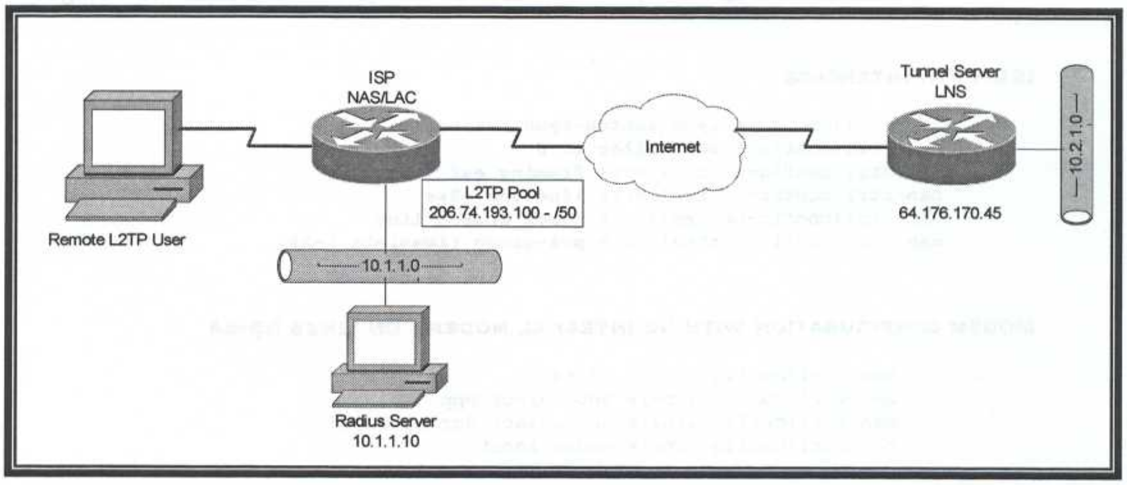
CHAPTER 21

LAYER 2 TUNNELING PROTOCOL

L2TP is a secure protocol used for connecting VPNs (Virtual Private Networks) over public lines such as the Internet. It is essentially a combination of two other secure communications protocols: PPTP and Cisco's Layer 2 Forwarding Protocol (L2F).

The layout for an L2TP VPN consists of a NAS and a tunnel server. The NAS is maintained by the ISP or provider. The NAS receives incoming calls for dial-in VPNs and places outgoing calls for dial-out VPNs. Normally when we think of a NAS we think of the access routers at a customer's site. The tunnel server terminates dial-in VPNs and initiates dial-out VPNs. This is usually maintained at the customer's site and can be managed by either the customer or the ISP. Typically, the customer will manage this device since this is the entrance point for VPN users into their private network.

Figure 21.1 *Typical L2TP network setup*



L2TP Configuration

Depending upon your role in an L2TP network, ISP or customer, your configurations may vary. The NAS is often referred to as the L2TP Access Concentrator (LAC). The tunnel server is often called the L2TP Network Server (LNS). All of our configurations will be based on NAS-initiated VPN's. NAS-initiated VPNs take place when users dial in to the ISP's NAS and establish a tunnel to the corporate network. From the NAS to the tunnel server, the connection is encrypted. However, the phone line connection between the client's PC and the ISP's NAS is not encrypted. This is generally not considered a high security risk, as is the Internet.

NAS/LAC Configuration

Step 1 Enable basic AAA commands for user authentication. Our example uses RADIUS since most dialup implementations use RADIUS.

```
nas-rtrl(config)# aaa new-model
nas-rtrl(config)# aaa authentication login ppp radius local
nas-rtrl(config)# radius-server host 10.1.1.10
nas-rtrl(config)# radius-server key cisco
```

Step 2 Configure the local pool of IP addresses.

```
nas-rtrl(config)# ip local pool l2tp-pool 172.16.1.10 172.16.1.100
```

Step 3 Configure the interface that accepts PPP calls. These examples show two typical interface types and their configurations. Consult your ISDN PRI circuit provider for exact configuration parameters. The following are the most common configurations.

ISDN PRI Interface

```
nas-rtrl(config)# isdn switch-type basic-ni
nas-rtrl(config)# controller t1 0
nas-rtrl(config-controller)# framing esf
nas-rtrl(config-controller)# linecode b8zs
nas-rtrl(config-controller)# clock source line
nas-rtrl(config-controller)# pri-group timeslots 1-24
```

Modem configuration with 32 internal modems on lines 33-64

```
nas-rtrl(config)# line 33 64
nas-rtrl(config-line)# autoselect ppp
nas-rtrl(config-line)# autoselect during-login
nas-rtrl(config-line)# modem inout
```

Step 1 Enable VPDN on the NAS.

```
nas-rtrl(config)# vpdn enable
```

Step 2 Configure the tunnel username and password. The username is the OTHER router's hostname or local name. The password must be the same on both sides.

```
nas-rtrl(config)# username lns-rtrl password secret
```

-OR-

Configure the VPDN group password and optional username. By default, the router will send its hostname as its username to the other router. If you want the router to send a username other than the hostname, use the optional command below.

```
nas-rtr1(config)# vpdn-group 1
nas-rtr1(config-vpdn)# l2tp tunnel password
```

The following command will send the username `nas` instead of `nas-rtr1`. This command is optional.

```
nas-rtr1(config-vpdn)# local name nas
```

Step 3 Configure the VPDN group parameters. The first command enables the NAS to accept incoming dial-up requests. The second command selects the protocol to be used (either l2tp, l2f, or any).

```
nas-rtr1(config-vpdn)# request-dialin
nas-rtr1(config-vpdn-req-in)# protocol l2tp
nas-rtr1(config-vpdn-req-in)#  domain name cisco.com
```

Step 4 Configure the IP address of the tunnel server that the NAS will connect to. The limit and priority commands are optional. These commands set a limit of 50 connections for this VPDN group and set the priority to 10.

```
nas-rtr1(config-vpdn)# initiate-to-ip 64.176.170.45 limit 50 priority 10
```

Tunnel Server/LNS Configuration

Step 1 Enable VPDN on the router.

```
lns-rtr1(config)# vpdn enable
```

Step 2 Configure the VPDN group.

```
lns-rtr1(config)# vpdn-group 1
```

Step 3 Accept dial-in connections from the NAS.

```
lns-rtr1(config-vpdn)# accept-dialin
```

Step 4 Configure the VPDN group parameters. The LNS has the additional command to specify the virtual access interface.

```
lns-rtr1(config-vpdn-acc-in)# protocol l2tp
lns-rtr1(config-vpdn-acc-in)# domain name cisco.com
lns-rtr1(config-vpdn-acc-in)# virtual-template 1
```

Step 5 Configure the hostname the tunnel should receive from the NAS, either the actual hostname of the NAS or the local name. This was explained in Step 4 of the NAS configuration.

```
lns-rtr1(config-vpdn)# terminate-from hostname nas-rtr1
```

Step 6 Create the local pool of IP addresses.

```
Lns-rtr1(config)# ip local pool l2tp-pool 10.2.1.100 10.2.1.150
```

Step 7 Create the virtual template interface. Configure an IP address (typically unnumbered using the internal interface), authentication types (CHAP or PAP), IP address pool for incoming PPP client connections, and the encapsulation type (PPP).

```
lns-rtr1(config)# interface virtual-template 1
lns-rtr1(config-if)# ip unnumbered fastethernet 0/0
lns-rtr1(config-if)# ppp authentication chap pap
lns-rtr1(config-if)# peer default ip address pool l2tp-pool
lns-rtr1(config-if)# encapsulation ppp
```

Typical Gotchas!

- Incorrect AAA configuration on NAS
- Access-dialin and request-dialin on the wrong routers
- Missing l2tp protocol command under access-dialin and request-dialin
- Misconfigured virtual template interface

CHAPTER 22

GRE TUNNELS AND IPSEC

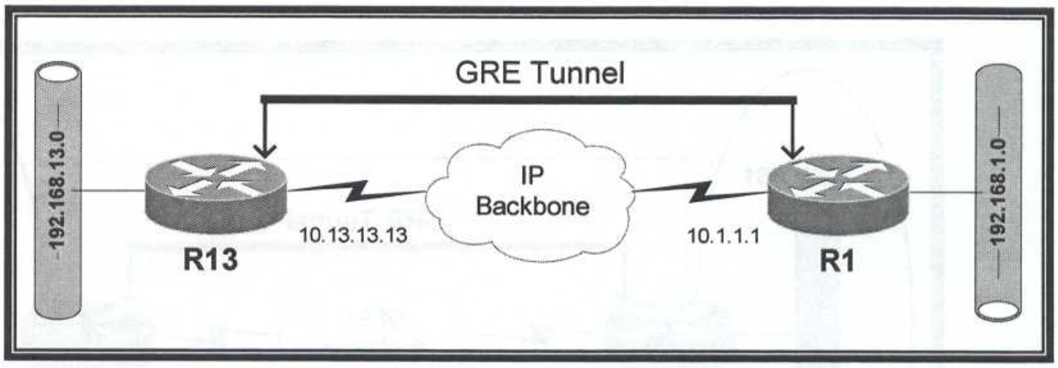
GRE OVERVIEW

Generic Routing Encapsulation (GRE) has several uses. GRE can be used to tunnel non-IP protocols such as IPX or AppleTalk, extend routing protocol domains, or create directly connected networks to support discontinuous networks over an IP backbone. It can also be used with IPSec to provide encrypted tunnels.

BASIC GRE CONFIGURATION

In Figure 22.1, we have a simple GRE topology. We can connect the two networks on each end of the IP cloud. We can use static routes or a routing protocol over the tunnel. Prior to configuring the GRE tunnel, make sure the tunnel endpoints, 10.13.13.13 and 10.1.1.1 in this example, can reach each other.

Figure 22.1 *Basic GRE topology*



Step 1 Create a Tunnel interface on each endpoint router.

```
r1(config)# interface tunnel0
```

```
r13(config)# interface tunnel0
```

Step 2 Specify a tunnel source and destination IP address. These IP addresses should be the physical addresses that are reachable via static routes or a routing protocol.

```
r1(config-if)# tunnel source 10.1.1.1
```

```
r1(config-if)# tunnel destination 10.13.13.13

r13(config-if)# tunnel source 10.13.13.13
r13(config-if)# tunnel destination 10.1.1.1
00:18:38: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed state
    to up
```

Step 3 Configure an IP address on each end of the tunnel. A tunnel is a point to point link, so treat it in a similar manner as you would a serial connection.

```
r1(config)# interface tunnel0
r1(config-if)# ip address 192.168.100.1 255.255.255.0

r13(config)# interface tunnel0
r13(config-if)# ip address 192.168.100.13 255.255.255.0
```

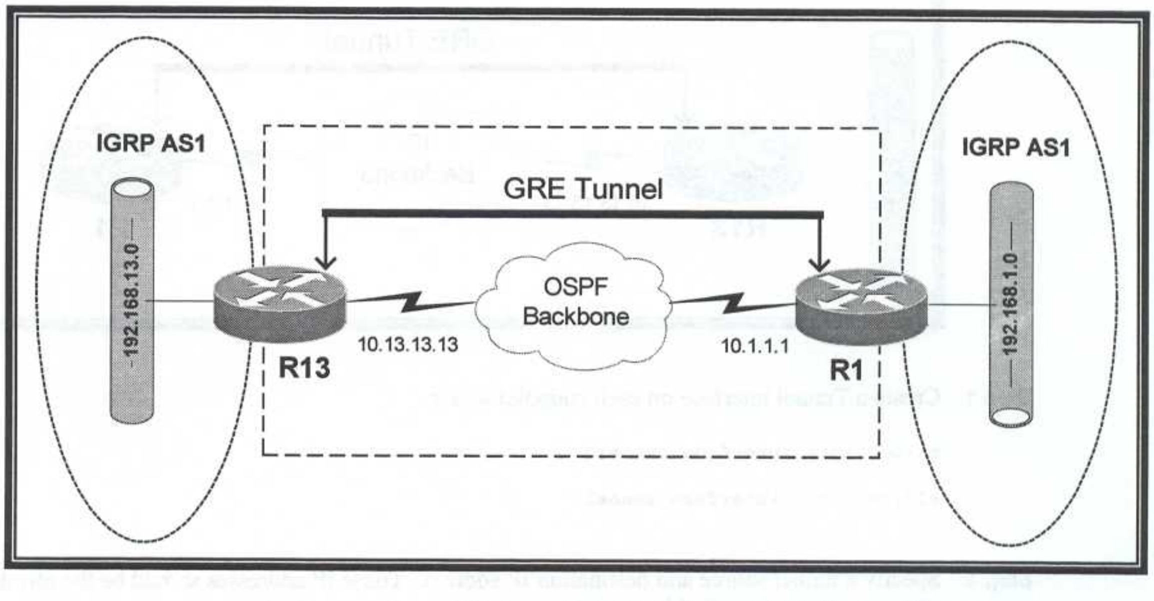
Step 4 Create a static route so that each LAN knows how to reach the other LAN via the tunnel. Alternatively, you can configure a routing protocol over this tunnel as explained in the next section.

```
r1(config)# ip route 192.168.13.0 255.255.255.0 tunnel0
r13(config)# ip route 192.168.1.0 255.255.255.0 tunnel0
```

GRE AND ROUTING PROTOCOLS

Tunnel interfaces can also run routing protocols in the same manner as any other point-to-point interface. In Figure 21-2, we have two IGRP networks that are running over an OSPF backbone. We can provide connectivity between the IGRP networks by running IGRP on the LAN interfaces and the tunnel.

Figure 22.2 *GRE and IP routing protocols*



```
r1(config)# router igrp 1
r1(config-router)# network 192.168.100.0
```

```
r1(config-router)# network 192.168.1.0

r13(config)# router igrp 1
r13(config-router)# network 192.168.100.0
r13(config-router)# network 192.168.13.0

r13# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    192.168.13.0/24 is directly connected, Ethernet0/0
     10.0.0.0/24 is subnetted, 2 subnets
O       10.1.1.0 [110/112] via 10.13.13.7, 00:00:04, Serial0/0
C       10.13.13.0 is directly connected, Serial0/0
I    192.168.1.0/24 [100/1161211] via 192.168.100.1, 00:00:04, Tunnel0
C    192.168.100.0/24 is directly connected, Tunnel0
```

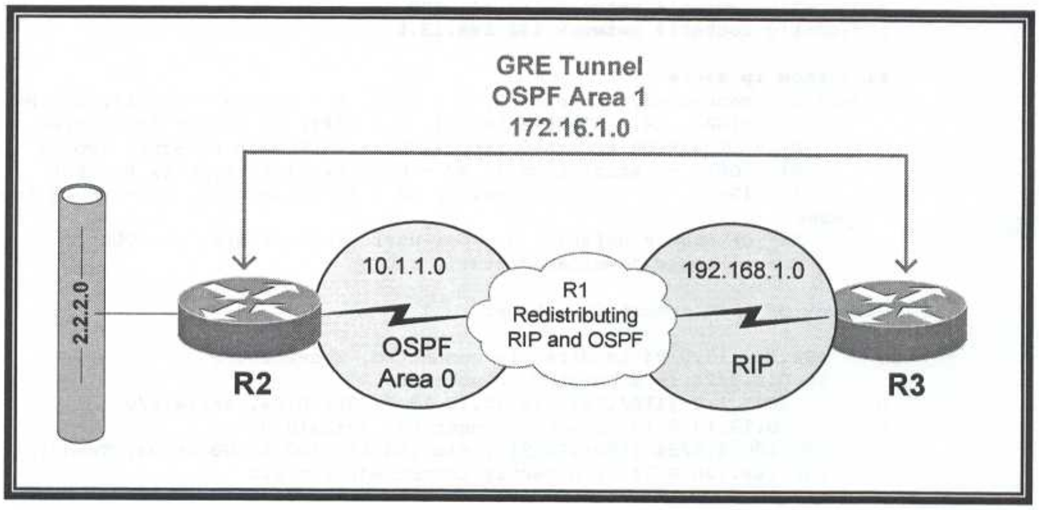
Avoiding Recursive routing

Beware of a routing situation where the best route to the tunnel destination is via the tunnel itself. Routing protocols that use hop count as their only metric are particularly susceptible. If you receive the following message you will have to correct the routing problem or your network will be down.

```
02:41:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to up
02:41:28: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive
    routing
02:41:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to down
02:42:29: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to up
02:42:38: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive
    routing
02:42:39: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to down
```

There are two ways to solve a recursive routing problem. The first way is to change the administrative distance for routes received over the tunnel. The second way is to filter routes received over the tunnel. In Figure 22.3, we have a layout that will generate a recursive routing error. The drawing below and relevant configs will be used for both solutions.

Figure 22.3 *GRE Recursive Routing*



```
R1
interface Serial0.1 point-to-point
 ip address 10.1.1.1 255.255.255.0
 frame-relay interface-dlci 102
!
interface Serial0.2 point-to-point
 ip address 192.168.1.1 255.255.255.0
 frame-relay interface-dlci 103
!
router ospf 1
 log-adjacency-changes
 redistribute rip metric 5 subnets
 network 10.1.1.1 0.0.0.0 area 0
!
router rip
 redistribute ospf 1 metric 1
 network 192.168.1.0
R2
interface Loopback0
 ip address 2.2.2.2 255.255.255.0
!
interface Tunnel0
 ip address 172.16.1.2 255.255.255.0
 tunnel source 10.1.1.2
 tunnel destination 192.168.1.3
!
interface Serial0.1 point-to-point
 ip address 10.1.1.2 255.255.255.0
 frame-relay interface-dlci 201
!
router ospf 1
 log-adjacency-changes
 network 2.2.2.2 0.0.0.0 area 0
 network 10.1.1.0 0.0.0.255 area 0
 network 172.16.1.2 0.0.0.0 area 1


R3
interface Tunnel0
 ip address 172.16.1.3 255.255.255.0
 tunnel source 192.168.1.3
 tunnel destination 10.1.1.2
```

```
!
interface Serial1.1 point-to-point
 ip address 192.168.1.3 255.255.255.0
 frame-relay interface-dlci 301
!
router ospf 1
 network 172.16.1.3 0.0.0.0 area 1
!
router rip
 network 192.168.1.0
```

SOLUTION 1: DISTANCE COMMAND

Once you configure the tunnel for OSPF it should attempt to come up. After OSPF is done loading, a few seconds later you will learn the tunnel endpoint (10.1.1.2) via OSPF as shown in the second show ip route command. This is the root cause of a recursive routing problem and will cause the tunnel to shut down. Once the tunnel goes down it will keep trying every minute. Your logs or console should fill up with these error messages as it will try forever until the tunnel is shut down or the problem is corrected.

```
r3#
08:24:44: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to up
08:24:50: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.1.2 on Tunnel0 from LOADING to
    FULL, Loading Done

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:19, Serial1.1
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
R       172.16.0.0/16 [120/1] via 192.168.1.1, 00:00:19, Serial1.1
C       172.16.1.0/24 is directly connected, Tunnel0
R    10.0.0.0/8 [120/1] via 192.168.1.1, 00:00:20, Serial1.1
C    192.168.1.0/24 is directly connected, Serial1.1

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    2.2.2.0/24 [110/11112] via 172.16.1.2, 00:00:03, Tunnel0
R       2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:29, Serial1.1
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
R       172.16.0.0/16 [120/1] via 192.168.1.1, 00:00:29, Serial1.1
C       172.16.1.0/24 is directly connected, Tunnel0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    10.1.1.0/24 [110/11175] via 172.16.1.2, 00:00:03, Tunnel0
C    192.168.1.0/24 is directly connected, Serial1.1
```

```
r3#

08:25:03: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive
    routing
08:25:04: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to down
08:25:04: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.1.2 on Tunnel0 from FULL to
    DOWN, Neighbor Down: Interface down or detached

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

R    2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:02, Serial1.1
R    172.16.0.0/16 [120/1] via 192.168.1.1, 00:00:31, Serial1.1
R    10.0.0.0/8 [120/1] via 192.168.1.1, 00:00:02, Serial1.1
C    192.168.1.0/24 is directly connected, Serial1.1
r3#
```

To begin to solve this problem we change the distance on the routes received from OSPF to be higher than RIP. Note that you need to use the Router ID (neighbor ID from the debugs). If a loopback happens to be the Nbr then that is what you want to use. Despite fixing the distance, the recursive routing still exists! In this particular scenario, our problem is caused by the more specific route that is generated by OSPF. Even though the RIP route is valid, distance only works if the routes have the same length. Don't forget that the router will always ignore a longer prefix. Once that happens, the router is back to using the route learned over the tunnel. Fortunately, there is a solution.

```
r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    2.2.2.0/24 [125/11112] via 172.16.1.2, 00:00:02, Tunnel0
R       2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:09, Serial1.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.1.0 is directly connected, Tunnel0
     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    10.1.1.0/24 [125/11175] via 172.16.1.2, 00:00:02, Tunnel0
R       10.0.0.0/8 [120/1] via 192.168.1.1, 00:00:09, Serial1.1
C    192.168.1.0/24 is directly connected, Serial1.1
r3#
08:43:44: %TUN-5-RECURDOWN: Tunnel0 temporarily disabled due to recursive
    routing
08:43:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface Tunnel0, changed
    state to down
08:43:45: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.1.2 on Tunnel0 from FULL to
    DOWN, Neighbor Down: Interface down or detached
```

The next step is to make the OSPF route be the same length as the RIP route. We do this by summarizing at the ABR. Then we clear routes and check R3 to make sure it is receiving the summary. Notice that the tunnel comes up and the routing is ok now. The 2.2.2.0 network is learned over the tunnel and the 10.0.0.0 network is still learned via RIP.

```
r2(config)# router ospf 1
r2(config-router)# area 0 range 10.0.0.0 255.0.0.0

r3# show ip ospf database

            OSPF Router with ID (192.168.1.3) (Process ID 1)

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
172.16.1.2      172.16.1.2      67          0x80000054 0x9FA1   2
192.168.1.3     192.168.1.3     66          0x8000007E 0x4524   2

                Summary Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
2.2.2.0         172.16.1.2      622         0x80000003 0xA1D2
10.0.0.0        172.16.1.2      101         0x80000001 0xE34F

                Summary ASB Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
192.168.1.1     172.16.1.2      622         0x80000004 0x8B42

                Type-5 AS External Link States

Link ID         ADV Router      Age         Seq#       Checksum Tag
192.168.1.0     192.168.1.1     1508        0x80000011 0x5B6E   0

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    2.2.2.0/24 [125/11112] via 172.16.1.2, 00:01:04, Tunnel0
R       2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:00, Serial1.1
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
R       172.16.0.0/16 [120/1] via 192.168.1.1, 00:01:41, Serial1.1
C       172.16.1.0/24 is directly connected, Tunnel0
R    10.0.0.0/8 [120/1] via 192.168.1.1, 00:00:00, Serial1.1
C    192.168.1.0/24 is directly connected, Serial1.1
```

Keep in mind that every scenario is different. Had the tunnel been on area 0 this solution would not work. You have to carefully review your routing tables to see exactly what is happening. Debugs can also be helpful, but you don't want to generate too much information either. You do not have a lot of spare time during the exam to look through several pages of debugs.

SOLUTION 2: FILTERING

Beginning with the starting configs (distance command on R3 and area range on R2 were removed), we are back to the recursive routing problem again. The filtering solution is much easier. Simply filter the route to the tunnel endpoint network and allow all other routes. R2's loopback is learned over the tunnel and the 10.1.1.0 network is still learned via RIP.

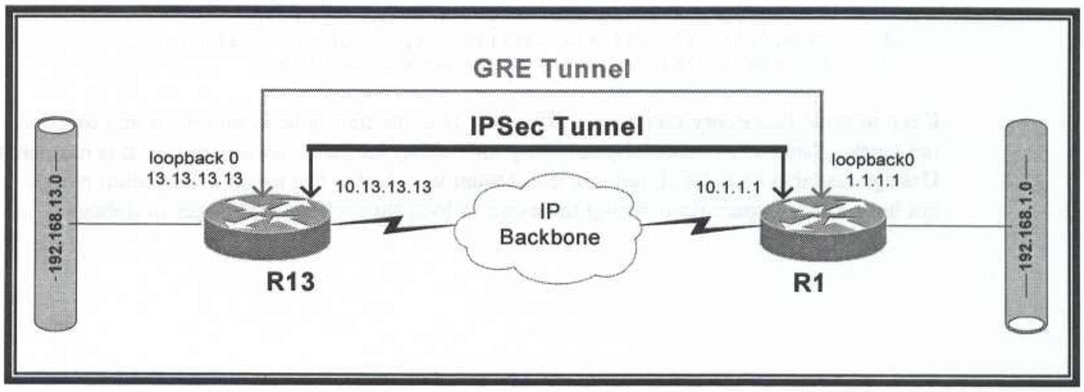
```
r3(config)# access-list 10 deny 10.1.1.0 0.0.0.255
r3(config)# access-list 10 permit any
r3(config)# router ospf 1
r3(config-router)# distribute-list 10 in tunnel0

r3# show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter
    area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     2.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA    2.2.2.0/24 [110/11112] via 172.16.1.2, 00:02:39, Tunnel0
R       2.0.0.0/8 [120/1] via 192.168.1.1, 00:00:22, Serial1.1
     172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
R       172.16.0.0/16 [120/1] via 192.168.1.1, 00:03:07, Serial1.1
C       172.16.1.0/24 is directly connected, Tunnel0
R    10.0.0.0/8 [120/1] via 192.168.1.1, 00:00:22, Serial1.1
C    192.168.1.0/24 is directly connected, Serial1.1
```

GRE OVER IPSEC

Why would you want to configure GRE with IPSec when VPNsalready provide a tunnel service? The answer is non-IP routing protocols cannot be encrypted using Cisco VPN's. The only way to encrypt non-IP protocols is to tunnel them with GRE and encrypt the entire GRE tunnel.

In Figure 21.4, we are going to create a Tunnel that will be protected by IPSec. The GRE tunnel endpoint should use a loopback address for redundancy. Make sure all IP addresses are reachable.

Figure 22.4 *GRE over IPSec*



Configuring GRE over IPSec is similar to a basic VPN with two notable differences. The first difference is that the **crypto map** is applied to both the physical interface and the tunnel. The second difference is the access list only encrypts the GRE traffic. It is always recommended to terminate GRE tunnels at loopback addresses for redundancy.

Router 1 Configuration

```
crypto isakmp policy 10
 hash md5
 authentication pre-share
crypto isakmp key cisco123 address 10.13.13.13
!
!
crypto ipsec transform-set weak esp-des esp-md5-hmac
 mode transport
!
crypto map GRE 10 ipsec-isakmp
 set peer 10.13.13.13
 set transform-set weak
 match address 101
!
interface Loopback0
 ip address 1.1.1.1 255.255.255.0
!
interface Tunnel0
 ip address 192.168.100.1 255.255.255.0
 tunnel source Loopback0
 tunnel destination 13.13.13.13
 crypto map GRE
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
!
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 crypto map GRE
!
ip route 0.0.0.0 0.0.0.0 10.1.1.7
ip route 192.168.13.0 255.255.255.0 Tunnel0
!
access-list 101 permit gre host 1.1.1.1 host 13.13.13.13
```

Router 13 Configuration

```
crypto isakmp policy 10
 hash md5
 authentication pre-share
crypto isakmp key cisco123 address 10.1.1.1
!
crypto ipsec transform-set weak esp-des esp-md5-hmac
 mode transport
!
crypto map GRE 10 ipsec-isakmp
 set peer 10.1.1.1
 set transform-set weak
 match address 101
!
interface Loopback0
 ip address 13.13.13.13 255.255.255.0
!
interface Tunnel0
 ip address 192.168.100.13 255.255.255.0
 tunnel source Loopback0
 tunnel destination 1.1.1.1
 crypto map GRE
!
interface Ethernet0/0
 ip address 192.168.13.13 255.255.255.0
!
interface Serial0/0
 ip address 10.13.13.13 255.255.255.0
 crypto map GRE
!
ip route 0.0.0.0 0.0.0.0 10.13.13.7
ip route 192.168.1.0 255.255.255.0 Tunnel0
!
access-list 101 permit gre host 13.13.13.13 host 1.1.1.1
```

Section V

AAA, IDS, and Network Management

CHAPTER 23

AUTHENTICATION, AUTHORIZATION, AND ACCOUNTING

Cisco routers, switches, and PIX firewalls all support AAA in some manner. All three can use TACACS+ and RADIUS for AAA, but only routers support a local database. A device that uses AAA services are called a Network Access Server (NAS).

- Authentication – Controls the authentication process through a username login and password prompt, challenge and response, and messaging support. Once a user sends his login and password a server (TACACS+ or RADIUS) may ask an additional challenge and response question to further verify the user. For example, the administrator can set a user's account to prompt for a predefined question and answer such as pet's name, mother's maiden name, last 4 digits of social security, etc. The messaging support can be used to inform users of specific notices such as password expiration.
- Authorization – Once a user is authenticated, they can be further restricted to what functions they can perform such as access control, session duration, protocol support, and so on. For example, the administrator can restrict the login on account *jdoe* to only 9am – 5pm.
- Accounting – information used for billing, auditing, and reporting is collected and sent to a server. This information can be used to monitor user activity, conduct security audits, or bill user's departments based on usage. The accounting records include user names, start and stop times, commands executed, and bandwidth utilized.

LOCAL AAA

The router's local database can be used for authentication. There is very little authorization that can be configured. The router has no ability to keep accounting records. Accounting must be handled by a server. PIX Firewalls and Catalyst switches do not have the ability to use local authentication. However, they can be configured to use either TACACS+ or RADIUS servers for AAA.

BASIC LOCAL AUTHENTICATION CONFIGURATION

Step 1 Enable Authentication, Authorization, and Accounting (AAA) with the **aaa new-model** global command.

```
r14(config)# aaa new-model
```

Step 2 Configure the method lists for authentication. The keyword "login" is used for Telnet and SSH login connections to the router itself. The keyword "default" is simply a default method list that is applied to all login capable interfaces and lines (VTY, console, AUX, etc.). The example below configures local authentication for all interfaces and lines.

```
r14(config)# aaa authentication login default local
```

Step 3 If using local authentication you must create at least one user account. Otherwise, you will not be able to login to the router.

```
r14(config)# username cisco password ccie
```

Telnet to the router and verify your local authentication is working prior to terminating your existing connection. It is also advisable not to save your configuration until you verify your new configuration is working properly. This way, if you do make a mistake and inadvertently lock yourself out of the router, a reboot will remove changes made since the last save. For that reason, it is highly suggested that you save your current configuration prior to configuring any AAA commands.

Named Method Lists

In the previous example we configured a default method list for logins to a router. Let's assume that the default is sufficient for all lines and interfaces except for our console port. We want the console port to use the line password instead of the local database. Since we can only have one default method list, we must create a named method list and apply it to the console port. The word "CONSOLE" is an arbitrary name used to identify the method list. It is helpful to use a descriptive name and use all caps to easily identify your method list names.

```
r14(config)# aaa authentication login CONSOLE line
r14(config)# line console 0
r14(config-line)# login authentication CONSOLE
```

Local Authorization Configuration

Authorization commands also use method lists. A default method list or named method list can be created.

Once a user is authenticated, the NAS can check if a user is authorized to perform the requested function. Exec authorization allows a user to enter the shell (exec) mode using either Telnet, SSH or other login transport. The specific commands a user is allowed to enter can also be controlled through command authorization. Network services such as PPP, can also use authorization to control PPP dialup users.

Exec Authorization

Step 1 Verify authentication is configured and working properly.

Step 2 Configure authorization for shell access. Shell access (exec mode) authorization requires that the user be authenticated first.

```
r14(config)# aaa authorization exec default local
```

Command Authorization

Step 1 Verify authentication is configured and working properly.

Step 2 Configure authorization to verify the user is allowed to enter specific commands. If a user requests a command that is privilege level 15, they must be authorized by the local database. The local database includes the default privilege levels for all commands as well as any modifications done with the **privilege exec level** command. The **privilege** command is explained later in this chapter.

```
r14(config)# aaa authorization commands 15 default local
```

Configuration Command Authorization

In the previous example, we enabled authorization for commands entered on the router. However, this authorization does NOT cover configuration commands. Notice the output of the **debug aaa authorization** after we entered a configuration command.

```
r14(config-if)# shut
23:25:09: AAA/AUTHOR: config command authorization not enabled
```

To enable configuration command authorization use the following command.

```
r14(config)# aaa authorization config-commands
```

Notice the **debug aaa authorization** output when we login and enter the same commands as the previous example. You can trace each command as shown in the highlighted debug output below.

```
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): Port='tty130' list=''
    service=CMD
23:26:38: AAA/AUTHOR/CMD: tty130 (1915216716) user='test'
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): send AV service=shell
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): send AV cmd=configure
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): send AV cmd-arg=terminal
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): send AV cmd-arg=<cr>
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): found list "default"
23:26:38: tty130 AAA/AUTHOR/CMD (1915216716): Method=LOCAL
23:26:38: AAA/AUTHOR (1915216716): Post authorization status = PASS_ADD
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): Port='tty130' list='' service=CMD
23:26:50: AAA/AUTHOR/CMD: tty130 (502889752) user='test'
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV service=shell
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV cmd=interface
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV cmd-arg=Serial
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV cmd-arg=0
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV cmd-arg=0
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): send AV cmd-arg=<cr>
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): found list "default"
23:26:50: tty130 AAA/AUTHOR/CMD (502889752): Method=LOCAL
23:26:50: AAA/AUTHOR (502889752): Post authorization status = PASS_ADD
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): Port='tty130' list=''
    service=CMD
23:27:02: AAA/AUTHOR/CMD: tty130 (3007555429) user='test'
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): send AV service=shell
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): send AV cmd=shutdown
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): send AV cmd-arg=<cr>
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): found list "default"
23:27:02: tty130 AAA/AUTHOR/CMD (3007555429): Method=LOCAL
23:27:02: AAA/AUTHOR (3007555429): Post authorization status = PASS_ADD
23:27:04: %LINK-5-CHANGED: Interface Serial0/0, changed state to
    administratively down
```

Terminal Access Controller Access Control System (TACACS) Overview

Cisco supports three different types of TACACS with TACACS+ being the most current and therefore preferred of the three types. Regular TACACS and extended TACACS are no longer recommended for use by Cisco.

TACACS+ allows for Cisco routers, switches, and PIX firewalls to have a central database of users, permissions, and accounting records. The Cisco Secure Access Control Server product has both TACACS+ and RADIUS capability.

Basic TACACS+ Configuration

Step 1 Enable AAA.

```
r14(config)# aaa new-model
```

Step 2 Configure the TACACS+ server(s). You are required to configure a host IP address and a key. The remaining parameters are optional.

```
r14(config)# tacacs host 10.1.1.100 ?
key                per-server encryption key (overrides default)
port               TCP port for TACACS+ server (default is 49)
single-connection  Multiplex all packets over a single tcp connection to
server (for CiscoSecure)
timeout            Time to wait for this TACACS server to reply (overrides
default)
<cr>
```

The `key` string is used to specify an encryption key for encrypting and decrypting all traffic between the NAS and the TACACS+ server. This key must be the same on the NAS and the TACACS+ server. Make sure your key is long enough since some vendors may require a minimum length. Entering a key is required for Cisco Secure ACS.

Use the `port` keyword to specify the TCP port number to be used when making connections to the TACACS+ server if you want to use a port other than the default (TCP port 49).

Use the `single-connection` keyword to specify single-connection. Instead of having the router open and close a TCP connection to the server each time it must communicate, the single-connection option maintains a single open connection between the NAS and the server. This allows the TACACS+ server to perform more efficiently and therefore handle a greater number of connections. This option requires CiscoSecure Release 1.0.1 or later.

Use the `timeout` value to specify the period of time (in seconds) the router will wait for a response from the server before it times out and declares an error.

TACACS Login Authentication

Once you have configured the TACACS+ host(s), you must tell AAA how and when to use the TACACS+ servers for authentication.

Step 1 Configure the method lists for authentication. Our example below will first attempt to use TACACS+ for authentication and if there is an error it will revert to local authentication. It is recommended to have at least one backup means to authenticate. If you use local authentication you have to configure usernames and passwords on the router itself. However, as long as the TACACS+ server is available and functioning properly the next method in the list will NOT be used. If the TACACS+ server responds with an ERROR or does not respond at all, only then will the next method will be used. The word "group" is required in order to tell the router to use either TACACS+ or RADIUS servers.

```
r14(config)# aaa authentication login default group tacacs+ local
```

Step 2 (Optional) If using local authentication as a backup method, you must create at least one user account.

```
r14(config)# username cisco password ccie
```

TACACS+ PPP AUTHENTICATION

Step 1 Configure AAA and the TACACS+ server.

```
r14(config)# aaa new-model
r14(config)# tacacs-server host 10.1.2.3 key secret
```

Step 2 Configure the method list for PPP to use TACACS+ and then local authentication in case the server is down or gets an error.

```
r14(config)# aaa authentication ppp DIALUP group tacacs+ local
```

Step 3 Apply the method list to an interface. If configured, the default method list will apply to all interfaces for that particular service, but a named method list must be specifically applied to an interface.

```
r14(config)# interface serial 0/0
r14(config-if)# ppp authentication chap DIALUP
```

TACACS+ AUTHORIZATION

Once a user is authenticated, you can limit what that user can do on the device. There are several different authorization types including exec, commands, network, etc. Assuming authentication and the TACACS+ server is configured properly, a user can be authorized to begin a shell (exec) session using either Telnet or SSH, begin a PPP network connection, or enter commands on the NAS.

LOGIN AUTHORIZATION EXAMPLE

Step 1 Verify authentication is configured and working properly using the TACACS+ server.

Step 2 Configure authorization for shell access.

```
r14(config)# aaa authorization exec default local if-authenticated
```

COMMANDS AUTHORIZATION EXAMPLE

Step 1 Verify authentication is configured and working properly using the TACACS+ server.

Step 2 Configure authorization to verify the user is allowed to enter specific commands.

```
r14(config)# aaa authorization commands 15 default local if-authenticated
```

PPP AUTHORIZATION EXAMPLE

Step 1 Verify authentication is configured and working properly using the TACACS+ server.

Step 2 Configure the default or named PPP method list. When a PPP connection is requested the user is first authenticated. If authorization is configured the user must be allowed to use the desired service. The network option is for services such as PPP, ARA, and SLIP.

```
r14(config)# aaa authorization network default group tacacs+
```

Step 3 Apply the method list to an interface.

```
r14(config)# interface serial 0/0
r14(config-if)# ppp authentication chap pap ccie
```

AAA SERVER GROUP CONFIGURATION TASKS

Scalable AAA services may involve configure a pool or group of servers that a NAS may contact for TACACS+ or RADIUS AAA services. Multiple servers can be contacted to respond to a specific AAA request.

Step 1 Configure a TACACS server(s) and any optional arguments such as encryption keys. The key can be configured as a global command if all servers will use the same key. Otherwise, they can be configured individually as show in the example below.

```
r14(config)# tacacs-server host 10.1.2.3 key code123
r14(config)# tacacs-server host 192.168.2.3 key secret123
```

Step 2 Configure the AAA group for TACACS.

```
r14(config)# aaa group server tacacs+ ACS
```

Step 3 Add servers to the AAA group. These servers must be configured with the `tacacs-server host` command before they can be added to a group.

```
r14(config-sg-tacacs)# server 10.1.2.3
r14(config-sg-tacacs)# server 192.168.2.3
```

TACACS+ ACCOUNTING

Accounting allows the administrator to monitor services users are accessing as well as the amount of resources and bandwidth they are using. By tracking the different services a network administrator can clearly see which services are used and for how long. This is essential for billing and/or security purposes.

To configure TACACS+ accounting the administrator must first configure basic AAA commands. AAA accounting can then be configured depending on which service to be tracked. There are 5 different types of accounting services: command, connection, exec, network, and system.

- Command accounting - provides information about the EXEC shell commands for a privilege level that are being executed on a NAS. Each command accounting record includes a list of the commands executed for that privilege level as well as the date and time each command was executed, and the user who executed it.
- Connection accounting - provides information about all outbound connections made from the NAS including Telnet, LAT, TN3270, PAD, and rlogin.
- EXEC accounting - provides information about user EXEC terminal sessions (user shells) on the network access server, including username, date, start and stop times, the access servers IP address, and (for dial-in users) the originating telephone number.
- Network accounting - provides information for all PPP, SLIP, or ARAP sessions including packet and byte counts.
- System accounting - provides information about all system-level events (for example, when the system reboots or when accounting is turned on or off).

TACACS+ Accounting Configuration

Step 1 Enable accounting and select which service you want to track (commands, connection, exec, network, resource, or system).

```
r14(config)# aaa accounting ?
  auth-proxy    For authentication proxy events.
  commands      For exec (shell) commands.
  connection    For outbound connections. (telnet, rlogin)
  exec          For starting an exec (shell).
  nested        When starting PPP from EXEC, generate NETWORK records
  before EXEC-STOP record.
  network       For network services. (PPP, SLIP, ARAP)
  resource      For resource events.
  send          Send records to accounting server.
  suppress      Do not generate accounting records for a specific type of
  user.
  system        For system events.
  update        Enable accounting update records.
```

Also, you must choose how you want the accounting records handled. If you want to minimize the accounting entries use stop-only; otherwise an entry is made to mark both the beginning and end of an accounting record.

```
r14(config)# aaa accounting network DIALUP ?
  none        No accounting.
  start-stop  Record start and stop without waiting
  stop-only   Record stop when service terminates.
  wait-start  Same as start-stop but wait for start-record commit.

r14(config)# aaa accounting network DIALUP start-stop group tacacs+
```

Step 2 (Optional) Apply the accounting configuration to a specific interface if using a named method list.

```
r14(config)# interface serial 0/0
r14(config-if)# ppp accounting DIALUP
```

Remote Authentication Dial-In User Service (RADIUS)

RADIUS is used to provide a centralized server to manage multiple network access points, typically for dial-up users. RADIUS has many similarities to TACACS+. Like TACACS+, RADIUS is supported by Cisco under its AAA commands including the authentication, authorization, and accounting services.

RADIUS Server Configuration

Before configuring your router or any NAS you should make sure your RADIUS server is functioning properly. There are several required and optional commands to configure RADIUS on a router.

Step 1 Enable AAA.

```
r14(config)# aaa new-model
```

Step 2 Configure the RADIUS server(s). There are several optional parameters after the IP address is entered as shown below.

```
r14(config)# radius-server host 192.168.1.10 ?
  acct-port          UDP port for RADIUS accounting server (default is 1646)
  alias              1-8 aliases for this server (max. 8)
  auth-port          UDP port for RADIUS authentication server (default is
    1645)
  key                per-server encryption key (overrides default)
  non-standard       Parse attributes that violate the RADIUS standard
  retransmit         Specify the number of retries to active server
(overrides default of 3 tries)
  timeout            Time to wait for this RADIUS server to reply (overrides
    default)
  <cr>
```

Step 3 Configure the RADIUS server(s) key. This key is used globally by all RADIUS servers. Specific keys can be configured on a per-server basis.

```
r14(config)# radius-server key ccie
```

RADIUS AAA Configuration

The commands for issuing AAA using RADIUS are basically the same as TACACS+ commands and will not be discussed in detail. The following example is a general configuration using RADIUS with the AAA command set. This example uses all three AAA features for incoming modem connections using PPP.

```
r14(config)# radius-server host 192.168.1.10
r14(config)# radius-server key thekey2radius

r14(config)# username root password 2secret

r14(config)# aaa authentication ppp DIALUP radius local
r14(config)# aaa authorization network radius local
r14(config)# aaa accounting network start-stop radius

r14(config-line)# interface group-async 1
r14(config-line)# encapsulation ppp
r14(config-line)# ppp authentication chap DIALUP
```

Vendor Proprietary RADIUS server

The following example is identical to the previous, except for the two commands that are necessary to connect to a vendor-proprietary RADIUS server.

```
r14(config)# radius-server host bigbird non-standard
r14(config)# radius-server key notciscopassw0rd
r14(config)# radius-server configure-nas
r14(config)# username root password 2secret
r14(config)# aaa authentication ppp dialups radius local
r14(config)# aaa authorization network radius local
r14(config)# aaa accounting network start-stop radius
r14(config)# aaa authentication login admins local
r14(config)# aaa authorization exec local
r14(config)# line 1 16
r14(config-line)# autoselect ppp
r14(config-line)# autoselect during-login
r14(config-line)# login authentication admins
r14(config-line)# interface group-async 1
r14(config-line)# encapsulation ppp
r14(config-line)# ppp authentication pap dialups
```

The `radius-server host <hostname> non-standard` command defines the name of the RADIUS server host and identifies that this RADIUS host uses a vendor-proprietary version of RADIUS.

The `radius-server configure-nas` command tells the Cisco router or access server to query the RADIUS server for static routes and IP pool definitions when the device first starts up. This command is optional and is only available with some RADIUS platforms.

Privilege Levels

Privilege levels are used by the router to determine if a user may enter a command. By default, a user is given a privilege level 1 after a successful login. Enable mode gives the highest privilege level; 15. The router can be configured to assign a new privilege level to specific commands. A router can also be configured to assign a user to a specific privilege level after a successful login.

To change a user's privilege level enter the command `username <name> privilege <level> password <string>`.

```
r14(config)# username User1 privilege 7 password cisco
```

In the previous example, we create a user with privilege level 7. This does us little benefit since by default, all Cisco router commands are either at level 1 or level 15. So, until we change the privilege level of one or more commands, we won't have accomplished anything different from the defaults. Here we set the privilege level of `show config` to level 7. Now User1 can enter all of the level 1 commands as well as `show config`. Remember that show run is normally a level 15 command. This is very common for NOC environments where we want to give administrators the privilege to do certain show commands, but do not want to give them full enable access to make changes.

Note Changing the privilege level of show run or write terminal will allow you to enter the command, but it show you a blank configuration unless you set the command to privilege level to 10 or higher.

You can also create multiple enable passwords for various privilege levels. For example, you can leave all your user accounts at the default level of 1. Then, assign multiple enable passwords for the various levels that you want to allow users to have. The example below creates two different enable passwords (in addition to the regular enable secret password at level 15) that can be given to network administrators or users of the router.

```
r14(config)# enable secret level 10 cisco2
r14(config)# enable secret level 7 cisco3
```

If you are asked to specifically allow a user only a single command, change the privilege level of that command to 0. Level 0 commands are used when you want to restrict a user to a limited command set. The issue with setting the privilege level for a command like **show ip route** is that all **show** and **show ip** commands are automatically set to the same privilege level that you set **show ip route**. The solution is to either set all show commands individually to their desired privilege level, which would be extremely tedious and time consuming, or set the **show ip route** command to level 0 so it does not affect the other commands in its subset.

```
r14(config)# privilege exec level 0 show ip route
```

Note that when you look in the configuration, it automatically applies the lower level **show** and **show ip** commands. You need to have the lower commands in order to get to show ip route. However, as you can see in the example below, you are still limited to only entering the **show ip route** command.

```
r14# show run
<edited for brevity>
!
privilege exec level 0 show ip route
privilege exec level 0 show ip
privilege exec level 0 show
!

r14> show ip ?
  route  IP routing table
```

By default, there are five commands associated with privilege level 0: disable, enable, exit, help, and logout. These commands can be changed to another security level. If you do not want users with a privilege level of 0 to even attempt to enter enable mode, change the security level to 1 or higher. Pay close attention to your user's level when they login to the router. If you set the enable command mode to level 7 and users login with the default of level 1, they will not be allowed to enter the enable command. Make sure this is the desired result.

Typical Gotchas!

- Configuring local authentication and not defining any user accounts
- Misconfigured authorization that locks you out of configuring the router
- TACACS or RADIUS key missing or incorrect on either the router or the ACS
- Privilege levels are for the exact command

CHAPTER 24

INTRUSION DETECTION

AS of this writing, the IDS Sensor and IDS module for the 6000/6500 Catalyst switch are not part of the CCIE Security lab. However, there are plans to add the IDS module in the future. This chapter explains how to configure IDS on a router and a PIX firewall.

IDS Configuration on a Router

Prior to configuring IDS on a router you must have an image that supports IDS. Make sure your hardware and software support IDS.

Standalone configuration (without an IDS management application)

Step 1 Set the notification type. The "log" keyword instructs the IDS software to send events to syslog.

```
router(config)# ip audit notify log
```

Step 2 Create the audit rules. Define what the action will be based on the signature type. The name "INTERNET" is arbitrary.

```
router(config)# ip audit name INTERNET info action alarm
router(config)# ip audit name INTERNET attack action alarm drop reset
```

info – informational signatures
attack – attack signatures

alarm – sends an alarm to the console, Director, CSPM, or to a syslog server
drop – drops the packet
reset – resets the TCP session

Step 3 Apply the audit rule to an interface.

```
router(config)# interface serial0
router(config-if)# ip audit INTERNET in
```

Configuration with an IDS management application

Step 1 Set the notification type. The "nr-director" keyword instructs the IDS software to send events to the Director or CSPM software.

```
router(config)# ip audit notify nr-director
```

Step 2 Create the audit rules. Define what the action will be based on the signature type. The name "INTERNET" is arbitrary.

```
router(config)# ip audit name INTERNET info action alarm
router(config)# ip audit name INTERNET attack action alarm drop reset
```

Step 3 Apply the audit rule to an interface.

```
router(config)# interface serial0
router(config-if)# ip audit INTERNET in
```

Step 4 Set the router's Postoffice settings. The hostid in our example is 15 and the orgid is the 11. By default, the hostid and orgid are 1.

```
router(config)# ip audit po local hostid 15 orgid 11
```

Step 5 Set the Director/CSPM Postoffice settings. The hostid in our example for the Director is 1. This command also specifies the remote Director/CSPM Postoffice IP address as well as its local IP address. This command can also change the communication port from the default UDP 45000. Preference, timeout, and application type can also be selected with this command. Everything after the local IP address has a default and only needs to be configured if the defaults are not desired.

```
router(config)# ip audit po remote hostid 1 orgid 11 rmtaddress
    192.168.1.10 localaddress 192.168.1.1 port 45000 preference 1 timeout
    10 application director
```

Additional IDS commands

ip audit smtp spam <number> - specifies the number of recipients on an email before it is considered as spam by the IDS. In our example, if an email has more than 20 recipients it is suspected as spam.

```
router(config)# ip audit smtp spam 20
```

ip audit po max-events <number> - specifies the maximum number of event notifications that are placed in the router's event queue. The default is 100. Each alarm uses 32KB of memory. Once the queue is full, the oldest alarms are deleted first.

```
router(config)# ip audit po max-events 200
```

ip audit po protected <IP address> to <IP address> - specifies that an IP address or network is on the protected network list.

```
router(config)# ip audit po protected 10.1.1.1 to 10.1.1.254
```

ip audit signature <sig-id> disable – specifies that a specific signature is disabled. By default, all 59 signatures are enabled.

```
router(config)# ip audit signature 3102 disable
```

ip audit signature <sig-id> list <access-list number> – specifies that a specific signature is disabled but only if the source IP address matches the standard access-list defined.

```
router(config)# access-list 10 deny host 10.1.1.45
router(config)# access-list 10 deny 172.16.1.0 0.0.0.255
router(config)# access-list 10 permit any
router(config)# ip audit signature 3102 list 10
```

IDS Verification

show ip audit all – displays all IDS information

show ip audit configuration – displays IDS configuration

show ip audit interfaces – displays all IDS interfaces

show ip audit name <name> – displays IDS configuration for a particular audit rule

show ip audit sessions – displays session information

show ip audit statistics – displays IDS statistics. This is the only information that is not included in the show ip audit all command.

```
router# show ip audit all
Event notification through syslog is enabled
Event notification through Net Director is disabled
Default action(s) for info signatures is alarm
Default action(s) for attack signatures is alarm
Default threshold of recipients for spam signature is 250
PostOffice:HostID:0 OrgID:0 Msg dropped:0
          :Curr Event Buf Size:0  Configured:100
Post Office is not enabled - No connections are active
Audit Rule Configuration
 Audit name INTERNET
    info actions alarm
    attack actions alarm drop reset
Interface Configuration
 Interface FastEthernet0/0
  Inbound IDS audit rule is INTERNET
    info actions alarm
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
 Interface FastEthernet0/1
  Inbound IDS audit rule is INTERNET
    info actions alarm
    attack actions alarm drop reset
  Outgoing IDS audit rule is not set
Established Sessions
 Session 827339D4 (192.168.1.22:3542)=>(209.189.89.243:119) tcp SIS_OPEN
 Session 8271746C (192.168.1.23:1610)=>(216.49.88.38:80) http SIS_OPEN

Terminating Sessions
 Session 82733490 (192.168.1.23:1608)=>(216.49.88.38:80) http SIS_CLOSING
 Session 82712220 (192.168.1.23:1609)=>(216.49.88.38:80) http SIS_CLOSING
 Session 8270C05C (192.168.1.23:1606)=>(216.49.88.38:80) http SIS_CLOSING
 Session 826FA538 (192.168.1.23:1607)=>(216.49.88.38:80) http SIS_CLOSING
 Session 82734EE8 (192.168.1.10:2632)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82736920 (192.168.1.10:2629)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82704D74 (192.168.1.10:2630)=>(208.47.125.33:80) http SIS_CLOSING
 Session 827091C4 (192.168.1.10:2631)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82711F90 (192.168.1.10:2624)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82735B54 (192.168.1.10:2625)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82736550 (192.168.1.10:2621)=>(208.47.125.33:80) http SIS_CLOSING
 Session 82733B48 (192.168.1.10:2622)=>(208.47.125.33:80) http SIS_CLOSING
```

```
router# show ip audit statistics
Signature audit statistics [process switch:fast switch]
  signature 2000 packets audited: [0:16]
  signature 2004 packets audited: [51:51]
Interfaces configured for audit 2
Session creations since subsystem startup or last reset 23154
Current session counts (estab/half-open/terminating) [2:9:13]
Maxever session counts (estab/half-open/terminating) [34:19:76]
Last session created 00:00:00
Last statistic reset never

Post Office is not enabled - No connections are active
```

PIX IDS CONFIGURATION

STANDALONE CONFIGURATION (WITHOUT AN IDS MANAGEMENT APPLICATION)

All IDS messages are sent to the configured logging buffer or Syslog.

Step 1 Create the audit rules. Define what the action will be based on the signature type. The PIX does not allow you to create one name for both audit rules.

```
pixfirewall(config)# ip audit name ATTACK1 info action alarm
pixfirewall(config)# ip audit name ATTACK2 attack action alarm drop reset
```

Step 2 Apply the audit rules to an interface.

```
pixfirewall(config)# ip audit interface outside ATTACK1
pixfirewall(config)# ip audit interface outside ATTACK2
```

TYPICAL GOTCHAS!

- Expecting the ip audit ? help command to work
- Not configuring logging
- Not applying the audit to an interface

CHAPTER 25

NETWORK MANAGEMENT

SECURE SHELL (SSH)

SSH is a secure protocol that encrypts a communication between a client and a server. SSH uses TCP port 22. SSH is mainly used as a replacement to telnet, rlogin, rexec, and rsh. Unlike telnet, usernames and passwords are sent encrypted not cleartext. In order to use SSH, the server will need to have an SSH daemon running to accept incoming SSH connections. Often administrators will open a port on the firewall to allow SSH to certain inside hosts. This may be a security risk if username and password combinations are weak. Even though the connection is secure, an attacker can still try various password cracking techniques just like telnet. Be cautious when opening SSH to the outside world. It is highly recommended that all network devices and servers both internally and on the Internet run SSH in place of telnet whenever possible.

CONFIGURING A CISCO ROUTER FOR SSH

In order to configure SSH on a router you need a DES or 3DES software image. SSH can be used in conjunction with AAA. Note that Cisco routers and PIX's only support SSH version 1.

Step 1 Configure a hostname

```
router(config)# hostname r13
```

Step 2 Configure an IP domain

```
r13(config)# ip domain-name netcginc.com
```

Step 3 Generate an RSA key and select the modulus. Note that some SSH clients cannot use keys larger than 1024. Check with your SSH client documentation.

```
r13(config)# crypto key generate rsa

The name for the keys will be: r13.netcginc.com
Choose the size of the key modulus in the range of 360 to 2048 for your
General Purpose Keys. Choosing a key modulus greater than 512 may take a
few minutes.

How many bits in the modulus [512]: 768
Generating RSA keys ...
[OK]
```

Step 4 Configure user authentication using either local, TACACS+, or RADIUS. In our example we use TACACS+ and then local.

```
r13(config)# aaa new-model
r13(config)# aaa authentication login default group tacacs+ local
r13(config)# username cisco password ccie
```

Step 5 Disable telnet on the VTY lines if desired. By default all protocols are permitted on a VTY (telnet, rlogin, etc).

```
r13(config)# line vty 0 4
r13(config-line)# transport input ssh
```

Step 6 (*Optional*) Configure commands such as timeouts and authentication retries. The default time-out value is 120 seconds. The default authentication-retries is 3.

```
r13(config)# ip ssh ?
  authentication-retries  Specify number of authentication retries
  time-out                Specify SSH time-out interval
```

Configuring a Cisco PIX for SSH

In order to configure SSH on a PIX you need version 5.2 or later with a DES or 3DES image. Unlike a router, SSH must be used in conjunction with either TACACS+ or RADIUS. The local authentication part is quite different from a router. On the PIX, the username is always "pix." There is no **username <name> password <string>** command. Also, if the AAA server is unavailable, local authentication will still work. To use only local authentication with SSH, omit all **aaa** and **aaa-server** commands.

Step 1 Set a Telnet and Enable password. Using **show** commands, you can see the passwords are automatically encrypted.

```
pixfirewall(config)# passwd cisco
pixfirewall(config)# enable password ccie

pixfirewall# show passwd
passwd xRrM7R12/DWq6mEK encrypted

pixfirewall# show enable
enable password yFzV1Ka232VO723 encrypted
```

Step 2 Set the allowed IP addresses for telnet. We are allowing all 10.0.0.0 networks to have telnet access. Note that this is only a list of allowed IP's. The Telnet user will still need to enter the passwords. Make sure you can telnet to the PIX before working on any AAA or SSH commands.

```
pixfirewall(config)# telnet 10.0.0.0 255.0.0.0 inside
```

Step 3 Configure AAA commands. The first command tells the PIX which protocol to use (TACACS+ or RADIUS). The second command configures the AAA server IP address and timeout. The name "pixauth" is arbitrary.

```
pixfirewall(config)# aaa-server pixauth protocol tacacs+
pixfirewall(config)# aaa-server pixauth host 10.1.1.119 ccie timeout 5
```

Step 4 Configure the connection(s) you want protected by AAA (telnet, serial, enable, or ssh). In this command the word "console" does not refer to the physical console port. PIX documentation calls the console port "serial." This may be confusing for those used to working with routers. The console refers to the command line of the PIX. This command basically says if you want to access the command line interface of the PIX (console) and your connection is telnet (or whichever is configured, i.e. telnet, serial, ssh, or enable) AAA authentication is in effect.

```
pixfirewall(config)# aaa authentication telnet console pixauth
```

Step 5 Configure a hostname (if not already configured)

```
pixfirewall(config)# hostname sf-pix1
```

Step 6 Configure an IP domain (if not already configured)

```
sf-pix1(config)# ip domain-name company.com
```

Step 7 Generate an RSA key and select the modulus.

```
sf-pix1(config)# ca generate rsa key 768
```

Step 8 Save the RSA key. Without saving the key, it will be lost on reboot. The **write memory** command does not save the key!

```
sf-pix1(config)# ca save all
```

Step 9 Set the allowed IP addresses for SSH.

```
sf-pix1(config)# ssh 10.0.0.0 255.0.0.0 inside
```

Step 10 Configure the SSH connection type to use AAA.

```
sf-pix1(config)# aaa authentication ssh console pixauth
```

Network Time Protocol (NTP)

NTP is used to synchronize the time of a computer client or server to another server or reference time source such as a radio, satellite receiver, or modem. NTP uses TCP port 123. There are two types of NTP servers: stratum 1 and stratum 2. Stratum 1 servers are primary and are considered the most accurate. Stratum 2 are very accurate and not as heavily utilized as stratum 1 servers. It is recommended that smaller environments (less than a couple hundred hosts) use stratum 2 NTP servers.

NTP is especially important for a secure environment because of time stamped system logging. With NTP configured the administrator can be fairly sure that the timestamps in the logs are correct. If an attacker penetrated the system and NTP wasn't running, the logs may not have the correct date and timestamps.

The perimeter router and/or firewall should be configured to only allow NTP from authorized stratum 1 or stratum 2 NTP server(s). Public and private keys are available to authenticate an NTP session. NTP can use either DES or MD5 authentication depending upon your NTP client and the server's capabilities. Cisco routers and switches use MD5. There are two reasons to use authentication. First, to ensure that only hosts that know the key can authenticate to the NTP server. Second, the host receiving the NTP information should know that the source of the information is valid. A host configured for NTP with public or private keys will not accept time information from an NTP server that it hasn't authenticated with.

Configuring Cisco routers for NTP

Cisco routers have the ability to use NTP to synchronize their clocks. They can use up to NTP version 3. They also have the ability to use authentication. Below is a simple example of an NTP client configuration using 2 different public NTP servers.

Step 1 Configure the NTP servers to synchronize with.

```
router(config)# ntp server 192.5.41.41
router(config)# ntp server 140.142.16.34
```

Step 2 Configure options such as authentication, preference, version, or source interface enter the commands after the NTP server's IP address.

```
router(config)# ntp server 192.5.41.41 ?
  key       Configure peer authentication key
  prefer    Prefer this peer when possible
  source    Interface for source address
  version   Configure NTP version
  <cr>
```

NTP Verification

Once NTP is configured verify the router has identified and synched with a clock. This may take a few minutes. The two **show ntp** commands give similar information and either command will tell you if the clock is synchronized.

```
router(config)# show ntp associations

address          ref clock     st  when  poll reach  delay  offset   disp
*~192.5.41.41       .USNO.      1   13    64     1    111.8  151.54  15875.
+~140.142.16.34     .USNO.      1    0    64     3    42.3   142.73  7942.4
master (synced), # master (unsynced), + selected, - candidate, ~ configured

router(config)# show ntp status
Clock is synchronized, stratum 2, reference is 192.5.41.41
nominal freq is 250.0000 Hz, actual freq is 249.9985 Hz, precision is 2**16
reference time is BEE369C5.1F531499 (13:06:29.122 pst Tue Jun 26 2001)
clock offset is 144.9341 msec, root delay is 111.83 msec
root dispersion is 16026.66 msec, peer dispersion is 15875.02 msec
```

Simple Network Management Protocol (SNMP)

SNMP is another method available to access and manage devices. SNMP can gather statistics or configure devices. Statistics can be viewed with get-request and get-next-request messages. Devices can be configured with set-request messages. Each of these SNMP messages has a community string that is a cleartext password sent in every packet between a management station and the Cisco device (which contains an SNMP agent). The SNMP community string is used to authenticate messages sent between the manager and agent. A network management station can be any system configured for SNMP. Typically, SNMP network management stations are commercial products such as HP Openview, Cisco Works, Tivoli, Micromuse Netcool, etc.

There are several security concerns with SNMP. First, SNMP strings are sent cleartext with SNMP version 1. Fortunately, Cisco devices can be configured to use SNMP version 2 which hashes the community strings using MD5.

SNMP community strings should be treated the same as your access and enable passwords. SNMP community strings can be subject to a brute force attack. If possible, you should limit your SNMP traffic to a specially created management VLAN or network only accessible by network administrators. If not, it is probably NOT good security practice to configure Read-write access.

It is also highly recommended to NOT use the default community names of public and private as these are common knowledge. Using these strings is like using "password" or "cisco" for your access passwords. Make your community strings as difficult as possible and even configure a different community string for each device.

String Types

SNMP devices have the ability to configure multiple community strings with different permissions. Typically, there is a Read-Only and a Read-Write string.

Read-Only Strings

The Read-only (RO) string is given the same permissions as a user with nonprivileged access. You can optionally use an access-list after the RO keyword to restrict SNMP access to a list of IP addresses. Our example below sets the Read-only string to public and will only accept SNMP messages from the 10.1.1.0 network.

```
router(config)# access-list 10 permit 10.1.1.0
router(config)# snmp-server community public RO 10
```

Read-Write Strings

The Read-Write (RW) string is given the same permissions as a user with privileged (enable level) access. It is generally not recommended to configure RW strings. Configure RW strings with caution.

```
router(config)# access-list 5 permit 172.16.10.0
router(config)# snmp-server community private RW 5
```

Logging

By default, system logs are only sent to the console port. Typically this console port is not monitored on a regular basis. If a terminal or PC is not connected to the console port no logs will be gathered. There are two solutions to this problem: configure logging to an internal buffer on the device or send the logs to a Syslog server.

Router Logging Configuration

To configure logs to be stored in memory on a router use the command **logging buffered**. The log will occupy space in memory so do not configure your log settings for more than your memory can handle. You can also set which level you want to log. If you select level 7 you will get all messages from 0-7. The level signifies the most sever messages you want to receive. In other words, if you select level 3 you will receive messages from levels 0 through 3. Messages in severity level 4 through 7 will not be logged. See below for an example of configuring the router to store 4096 bytes of messages and only log severity level 0 – 3.

```
router(config)# logging buffered ?
  <0-7>              Logging severity level
  <4096-2147483647>  Logging buffer size
  alerts             Immediate action needed          (severity=1)
  critical           Critical conditions              (severity=2)
  debugging          Debugging messages               (severity=7)
  emergencies        System is unusable               (severity=0)
  errors             Error conditions                 (severity=3)
  informational      Informational messages           (severity=6)
```

```
    notifications       Normal but significant conditions (severity=5)
    warnings            Warning conditions                (severity=4)
    <cr>

  router(config)# logging buffered 4096
  router(config)# logging buffered 3
```

Having the logs stored in a router's memory has three problems. First, if the router is reloaded the logs are lost. Second, if an attacker gains access to your router with privilege mode access they can erase your logs and you cannot recover them. Third, this is not very scalable in a large environment. The solution to both of these problems is to use a Syslog server to capture all logs. See below for an example to configure logs to be sent to a Syslog server with the IP address 10.1.1.100. We want to log all severity level messages.

```
  router(config)# logging 10.1.1.100
  router(config)# logging trap 7
```

PIX Logging

The PIX firewall sends syslog messages for the following events.

Security – denied TCP connections and dropped UDP packets
Resources – connection and/or translation slots running low
System – console and telnet logins; firewall reboots
Accounting – bytes transferred per second

By default, the syslog messages are sent only to the console. If no one is watching the console of the PIX the messages will not be stored anywhere! This is a very poor security practice. Syslog messages can be sent to the console, internal buffer, or a syslog server.

Messages sent only to the console are usually not sufficient for most security implementations. The console is not scalable because messages are not saved anywhere. They are displayed on the monitor of the attached terminal or PC. If the console connection is lost no messages will be received. Typically, most terminals or PC terminal software will have a limited buffer to store these messages. If there is a lot of messages being generated it may be only a matter of minutes before messages scroll of the screen and are no longer available. To reduce the number of messages there are two options. Note that these two options can also be used with the logging buffered command discussed in the next section.

Change the logging level to something less than debugging. The following are the logging levels. The default is level 7 which is all messages.

0 – emergencies
1 – alerts
2 – critical
3 – errors
4 – warnings
5 – notifications
6 – informational
7 – debugging

To change the level, use the following command. In our example, we are only allowing the PIX to log warning level messages are above.

```
pixfirewall(config)# logging console 4
```

Reducing PIX Log Messages

Do not allow the PIX to log specific messages. The **logging message <syslog_id>** command can be used to allow or disallow specific messages. When you enter the show log command it will list all the events. The first 6-digit number in front of the message is the *syslog_id*. Notice our show log and the syslog_id of 106023. All messages are allowed until specifically denied. The only message that cannot be denied is the *PIX Startup begin* message.

```
pixfirewall# show log
Syslog logging: enabled
    Timestamp logging: enabled
    Standby logging: disabled
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level warnings, 8517895 messages logged
    Trap logging: disabled
    History logging: disabled
106023: Deny icmp src outside:67.151.16.3 dst inside:26.18.11.1 (type 3, code
    1) by access-group "inbound"
106023: Deny icmp src outside:65.1.74.25 dst inside:26.18.11.1 (type 11, code
    0) by access-group "inbound"
```

If we wanted to stop the PIX from logging this message type we would use the following command.

```
pixfirewall(config)# no logging message 106023
```

PIX Logging Buffer Configuration

The internal buffer requires minimal configuration and allows the PIX to store up to 100 messages. The internal buffer uses the same commands as the console to reduce messages. Unlike console messages, the PIX must be configured to log messages to the buffer. The PIX does not log to the buffer by default. The following command enables logging to the buffer for all error level messages and higher.

```
pixfirewall(config)# logging buffered 3
```

PIX Syslog Configuration

The Syslog host configuration is the most scalable, but requires additional configuration. The following steps are needed to setup the PIX to send Syslog messages.

Step 1 Ensure the Syslog server is configured to receive messages.

Step 2 Configure a host to receive the messages. Our syslog server's IP address is 192.168.1.200 and can be found on the inside interface. We are using the standard UDP port 514. However, the UDP port can be changed to any value from 1025 through 65535. You can also specify TCP port 1470 if you are using the PIX Firewall Syslog Server software. The PIX can be configured to send messages to different servers, but you can only use one UDP or TCP protocol for each server. You cannot configure the same server for both UDP and TCP.

```
pixfirewall(config)# logging host inside 192.168.1.200
pixfirewall(config)# logging host inside 10.1.1.15 TCP/1470
```

Step 3 Set the logging trap level. This is the level discussed in the previous section. We are setting the level to debugging in our example.

```
pixfirewall(config)# logging trap 7
```

Step 4 *(Optional)* Change the default facility if desired. There are 8 facilities from Local0 to Local7. Local0 is expressed with the number 16 and Local7 with the number 23. The default facility for the PIX is local4 corresponding to facility 20. This command is only necessary to change from the default facility of 20.

```
pixfirewall(config)# logging facility 21
```

Step 5 *(Optional)* Enable timestamps if desired.

```
pixfirewall(config)# logging timestamp
```

Step 6 Enable logging. This command is necessary for the PIX to send messages to the syslog server.

```
pixfirewall(config)# logging on
```

Step 7 Check to make sure that all the settings are correct. This command will also show logs sent to the buffer.

```
pixfirewall# show log
Syslog logging: enabled
    Timestamp logging: enabled
    Standby logging: disabled
    Console logging: disabled
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level warnings, 8546578 messages logged
    Trap logging: level emergencies, facility 21, 0 messages logged
        Logging to inside 192.168.1.200
        Logging to inside 10.1.1.15 tcp/1470
    History logging: disabled
```

Typical Gotchas!

- Not saving RSA keys on the PIX
- NTP authentication keys not the same
- Clocks between two NTP devices are too far apart to synchronize
- AAA not configured on the router to support SSH