Let's say that you've been assigned a Class B network address of 180.10.0.0. To subnet this network, you will have to steal bits from the third octet. You have determined that you want to create six subnets. Figure 10.11 walks you through the process of creating the subnets and creating the new subnet mask.

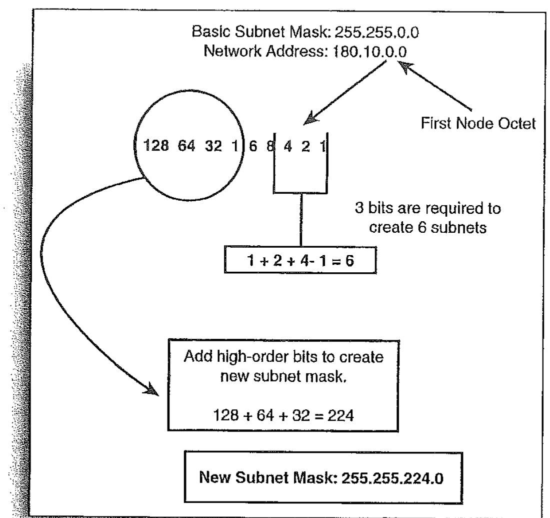


FIGURE 10.11
Determine the lower order bits needed to create the subnets and then add the same number of higher order bits to create the subnet mask.

The new subnet mask for the network would be 255.255.224.0 (see Figure 10.12). To figure out the range of IP addresses in each of the six subnets, you use the lowest of the high-order bits that were added to determine the new subnet mask number for the third octet. This would be 32 (again, taken from Figure 10.12). So, the first address in the first subnet would be 180.10.32.1 (180.10.32.0 is reserved as the subnetwork address and so cannot be used as a node address). To come up with the starting IP address of the second subnet, add 32 to the third octet (64). The second subnet would start with 180.10.64.1. Table 10.5 shows the ranges for the six subnets created from this Class B network address.

Table 10.5 IP Address Ranges for Class B

Subnet #	Start Address	End Address
1	180.10.32.1	180.10.63.254
2	180.10.64.1	180.10.95.254
3	180.10.96.1	180.10.127.254
4	180.10.128.1	180.10.159.254
5	180.10.160.1	180.10.191.254
6	180.10.192.1	180.10.223.254

Because you took 3 bits to create your subnets, you are left with 13 bits for nodes. So, 2^{13}-2= 8190. That's 8190 IP addresses available per subnet.

Class C Subnetting

Class C subnetting is a little more problematic than Class A and B networks because you only have one octet to steal bits from to create your subnets. Class C networks are also small to begin with (only 254 IP addresses are available), so creating more than just a few subnets will leave you with a very small number of node addresses available in each subnet.

Let's walk through an example that allows us to examine the idiosyncrasies of Class C subnetting. The network address is 200.10.44.0. One octet is available for node addresses (the fourth octet). This is also the octet that you must borrow bits from to create your subnets.

You will divide the Class C network into two subnets. To create the two subnets you must borrow the first two lower order bits that have the decimal value of 1 and 2 (1+2-1=2 subnets). You then move to the other end of the decimal bit values and use the first 2 high-order bits (because you borrowed 2 bits for the subnets) to create the new subnet mask for the network. The two high-order bits are 128 and 64. Add them together and you get 192. So the new subnet mask for the network is 255.255.255.192.

Figure 10.12 summarizes the steps that were followed to create the new network subnet mask by borrowing the appropriate number of bits to create 2 subnets.

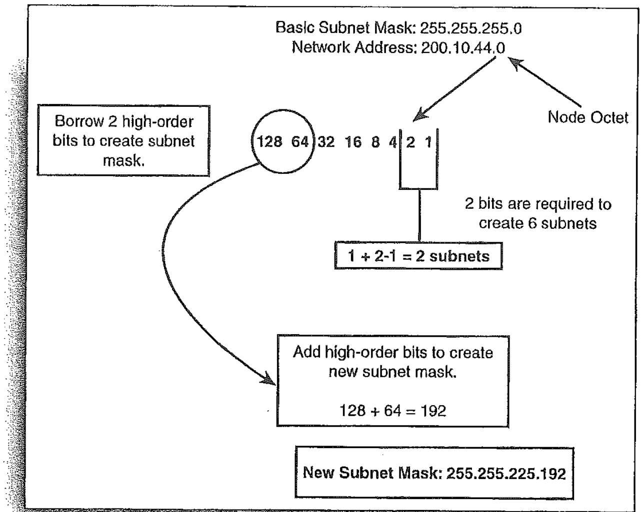


FIGURE 10.12
Use the number of lower order bits used to create the appropriate number of subnets and take the same number of high-order bits to create the subnet mask.

Now you need to figure out the range of IP addresses that will be available in the two subnets. The lowest of the high-order bits used to create the new subnet mask was 64, which becomes the increment for the subnet ranges. So, using what you learned when creating Class A and Class B subnets, you would assume that the start address of the first subnet would be 200.10.44.64. However, remember that an address in the range must be reserved as the subnetwork address. Because you are working with only one octet, the first usable address in the range of IP addresses for the subnet must be reserved as the subnetwork address. So, 200.10.44.64 is reserved for the subnet address.

That means that the beginning of the range of IP addresses in the first subnet that you can use for node addresses begins with 200.10.44.65. And the next subnet, which begins with 200.10.44.128 (you add the increment to itself to get the start of the next subnet range) also reserves the first address (200.10.44.128) as the subnetwork address (it identifies the subnet as a separate entity on the whole network). So the second subnet range of addresses that can be used for nodes begins with 200.10.44.129.

A name is just a name

I've been referring to the address provided by your ISP (such as 200.10.44.0) as the network address. This is also sometimes referred to as the major network address. And I've been identifying the address reserved for the subnet as the subnetwork or subnet address. In cases where the network address is referred to as the major network address, the subnetwork may be referred to as the network address. Just remember that the address you procure from InterNIC or your ISP is the network or major network address and the subnet addresses you create are subnetwork or network addresses.

Calculating available node addresses

To quickly calculate the number of IP addresses that would be available for each of our Class C subnets use the formula $2^{[\text{bits available for node addresses}]}$ minus 2. In our case this would be $2^6 - 2 = 62$. You have 2 subnets so 62x2=124.

Table 10.6 shows the ranges for the two Class C subnets and also shows addresses such as the subnetwork address that cannot be used for node addressing.

Table 10.6 IP Address Ranges for Class C Subnets (2)

Subnet	Subnetwork Address	Start Address	End Address	Broadcast Address
1	200.10.44.64	200.10.44.65	200.10.44.126	200.10.44.127
2	200.10.44.128	200.10.44.129	200.10.44.190	200.10.44.191

The big problem with subnetting a Class C network is that you lost a lot of normally usable IP addresses. You lost 2 addresses in each subnet, one for the subnetwork address, and one for the broadcast address. You also lost all the addresses that come before 200.10.44.64. That means you lose 200.10.44.1 through 200.10.44.63. That's quite a few addresses, especially when you don't get that many addresses with a Class C anyway.

Understanding Subnet 0

There is a way to "cheat" and use these lost addresses for your network nodes (in our case addresses 200.10.44.2 through 200.10.44.62-200.10.44.1 is reserved for the subnetwork address and 200.10.44.63 would be the broadcast address). These "lost" addresses are referred to as subnet 0 and normally cannot be used. However, you can configure your router to take advantage of the subnet 0 IP addresses: type the `ip subnet-zero` command at the config prompt and then press Enter (this is a global configuration command, so you don't have to enter it for any particular router interface).

Using subnet 0 means that only 1 bit needs to be stolen to create subnet 0 and subnet 1. So, the subnet mask would now be 255.255.255.128 (only 1 high-order bit is used to create the new subnet mask). The range of IP addresses for the two subnets would be 200.10.44.1-200.10.44.126 (200.10.44.127 is the broadcast address) for subnet 0 and 200.10.44.129-200.10.44.254 (200.10.44.128 is the subnetwork number and 200.10.44.255 is the broadcast address) for subnet 1.

Because using subnet 0 makes the calculation of subnets a little more difficult (when compared to Class A or B), Table 10.7 provides a summary of the fourth octet numbers that would be available for each subnet when a Class C network is subnetted with subnet 0 used as a valid subnet. Values are provided for 2, 4, and 8 subnets on the Class C network.

The big thing to remember when using subnet 0 is that you don't subtract 1 from the low-order bits when you determine the number of bits you must steal to create the required number of subnets.

Table 10.7 IP Address Ranges for Class C Subnets Using Subnet 0

# of Subnets	Subnet Mask	Start Address	End Address	Broadcast Address
2	255.255.255.128	x.x.x.1	x.x.x.126	x.x.x.127
		x.x.x.129	x.x.x.254	x.x.x.255
4	255.255.255.192	x.x.x.1	x.x.x.62	x.x.x.63
		x.x.x.65	x.x.x.126	x.x.x.127
		x.x.x.129	x.x.x.190	x.x.x.191
		x.x.x.193	x.x.x.254	x.x.x.255
8	255.255.255.224	x.x.x.1	x.x.x.30	x.x.x.31
		x.x.x.33	x.x.x.62	x.x.x.63
		x.x.x.65	x.x.x.94	x.x.x.95
		x.x.x.97	x.x.x.126	x.x.x.127
		x.x.x.129	x.x.x.158	x.x.x.159
		x.x.x.161	x.x.x.190	x.x.x.191
		x.x.x.193	x.x.x.222	x.x.x.223
		x.x.x.225	x.x.x.254	x.x.x.255

A Final Word on Subnetting

On any network that uses internetworking connectivity strategies, you will most likely face the issue of dividing a particular IP network into a group of subnets. And understanding the simple math presented in this chapter will make it very easy for you to create subnets on any class of network; however, sometimes it can be even simpler to just look up the information on a chart.

Table 10.8 provides a summary of the subnet mask and the number of hosts available when you divide a Class A network into a particular number of subnets (subnet 0 has not been allowed). Table 10.9 provides the same information for Class B networks (subnet 0 has not been allowed).

Table 10.8 Class A Subnetting

# Of Subnets	Bits Used	Subnet Mask	Hosts/Subnet
2	2	255.192.0.0	4,194,302
6	3	255.224.0.0	2,097,150
14	4	255.240.0.0	1,048,574
30	5	255.248.0.0	524,286
62	6	255.252.0.0	262,142
126	7	255.254.0.0	131,070
254	8	255.255.0.0	65,534

Table 10.9 Class B Subnetting

# Of Subnets	Bits Used	Subnet Mask	Hosts/Subnet
2	2	255.255.192.0	16,382
6	3	255.255.224.0	8,190
14	4	255.255.240.0	4,094
30	5	255.255.248.0	2,046
62	6	255.255.252.0	1,022
126	7	255.255.254.0	510
254	8	255.255.255.0	254

chapter

11

Configuring IP Routing

- Configuring Router Interfaces
- Configuring a Routing Protocol
- Dynamic Routing Versus StaticRouting
- Using Telnet

Configuring Router Interfaces

As you've already heard several times in this book, TCP/IP is the de facto network protocol for the networks of the world (due to the Internet explosion—everyone wants to be part of this planetwide network). It is a routable and robust network protocol stack. You learned all about IP addresses and IP subnetting in Chapter 10, "TCP/IP Primer." Now, you can take some of the concepts learned in that chapter and apply them directly to router configurations.

Routing IP on an internetwork requires that you complete two main tasks: configure LAN and WAN interfaces with the correct IP and subnet mask information, and then enable an IP routing protocol on your router or routers. (IP routing is automatically enabled on the router in contrast to IPX and AppleTalk, which aren't.) When routing IP, you have more than one choice for your routing protocol (such as RIP versus IGRP).

Let's walk through the steps of configuring LAN interfaces on a router first and apply some of the information that you picked up on IP subnetting in Chapter 10. For example, assume your example network is a Class B network with the network address 130.10.0.0. You will create 6 subnets on this network. The new subnet mask for the network would be 255.255.224.0.

Table 11.1 provides the range of IP addresses for the 6 subnets.

Table 11.1 IP Address Ranges for 6 Subnets on 130.10.0.0

Subnet #	Start Address	End Address
1	130.10.32.1	130.10.63.254
2	130.10.64.1	130.10.95.254
3	130.10.96.1	130.10.127.254
4	130.10.128.1	130.10.159.254
5	130.10.160.1	130.10.191.254
6	130.10.192.1	130.10.223.254

Figure 11.1 shows a diagram of a portion of a company internetwork. IP addresses (from our range in Table 11.1) have been assigned to the router interfaces on each of the routers. This figure will help provide some context to the IOS commands that you are going to work with in this chapter.

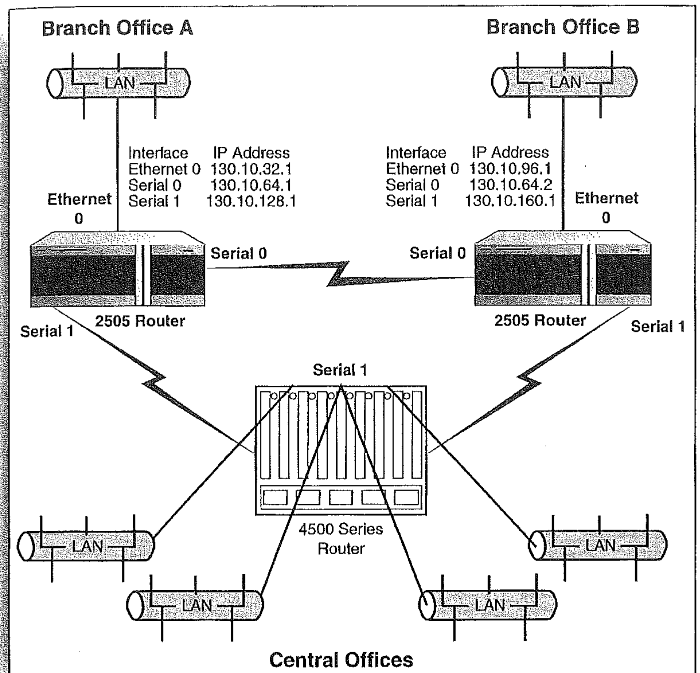


FIGURE 11.1
Two remote sites connected to a central office. IP addressing provided for remote sites.

You will configure the 2505 router at the Branch A location. This means that the router (which has three interfaces, one Ethernet, and two serial) must have each interface configured with a different IP address that is in a different subnet range. Table 11.2 lists the IP addresses (also shown in Figure 11.1) that you will use to configure this router. You will learn about configuring LAN interfaces (such as Ethernet ports) in the next section, "LAN Interfaces" and WAN interfaces in the section after that, "WAN Interfaces."

Table 11.2 IP Addresses for 2505 Router Interfaces

Interface	IP Address
Ethernet 0	130.10.32.1
Serial 0	130.10.64.1
Serial 1	130.10.128.1

SEE ALSO

➢ *For an overview of IP routing protocols such as RIP and IGRP, see page 93.*

LAN Interfaces

LAN interfaces, such as Ethernet ports or Token Ring ports, will b the connection point between the router and a local area network. The number of subnets at a particular location will dictate the num ber of LAN interfaces required on the router (if only one router is used).

Each of these LAN interfaces will be on a separate subnet. The sir plest way to assign IP addresses to a LAN interface is to use the fi IP address available in the address range of the subnet that the int face will connect to.

Configuring IP addressing for a LAN interface

1. At the Privileged prompt type `config t`, and then press Enter. You are placed in the Global Configuration mode.
2. To configure a particular LAN interface, type the name of th interface at the prompt, such as `interface ethernet 0`. Then press **Enter**. The prompt changes to the `config-if` mode.
3. Now you can enter the `ip address` command followed by the address for the interface and the subnet mask for the networ this example, the command would be `ip address 130.10.32.` `255.255.224.0` (see Figure 11.2). Press **Enter** to complete the command.
4. To end the configuration of the interface, press **Ctrl+Z**.
5. Press **Enter** again to return to the privileged prompt.

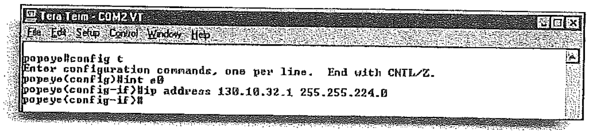


FIGURE 11.2
Individual LAN interfaces must be configured with an IP address and subnet mask.

You can quickly check the configuration parameters for a LAN port using the `show ip interface` command. For example, to see the IP addressing for Ethernet 0, you would type `show ip interface e0` and then press **Enter**. Figure 11.3 shows the results of this command on our 2505 router.

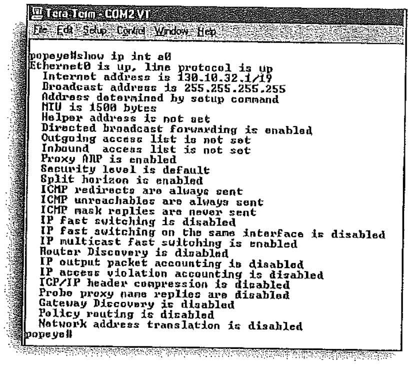


FIGURE 11.3
Check the IP addressing for an interface with the `show ip interface` command.

If you look at the IP information provided in Figure 11.3, the IP address reads as 130.10.32.1/19, and no subnet mask information is provided. You entered 130.10.32.1 as the IP address for the interface in the previous set of steps. So, what does the /19 mean? Actually, this is the router's way of telling you the subnet mask.

Show all interface IP addressing

If you type the `show ip interface` command and don't specify a particular router interface, the IP addressing of all the interfaces on the router will be displayed.

The 19 is the number of bits that are used for network addressing plus the number of bits used to create the subnets on this network. Normally, a Class B network uses two octets (16 bits) to define the network number for the network: in this case 19–16=3. This shows you the number of bits stolen for subnetting. If you take the first three high-order bits and add them (128+64+32), you get 224, which tells you that the subnet mask is 255.255.224.0.

Saving your router configuration

When you make changes to your router's configuration, you will want to save the configuration changes from RAM to NVRAM. This makes the currently running configuration file the startup configuration if the router is rebooted or powered back on after a power failure. At the privileged prompt, type `copy running-config startup-config`, and then press **Enter**. The configuration will be built and saved to NVRAM.

Whenever you see notation like the /19, just take that number and subtract the number of bits that are normally used for the class of network that you are working with. This always gives you the subnet bits, which can then be used to quickly calculate the subnet mask.

WAN Interfaces

WAN interfaces can be configured with IP addresses exactly in the same way that you configure LAN interfaces. To configure a serial 0 interface on a router, you would complete the following steps.

Configuring IP addressing for a serial interface

1. At the Privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To configure a particular LAN interface, type the name of the interface at the prompt, such as interface serial 0. Then press **Enter**. The prompt changes to the `config-if` mode.
3. Now you can enter the IP address command followed by the IP address for the interface and the subnet mask for the network. In this example, the command would be `ip address 130.10.64.1 255.255.224.0` (see Figure 11.4). Press **Enter** to complete the command.

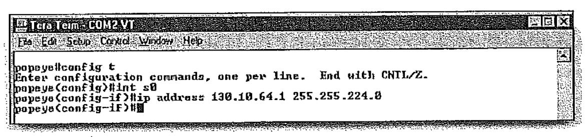


FIGURE 11.4
Individual WAN interfaces must be configured with an IP address and subnet mask.

4. To end the configuration of the interface, press **Ctrl+Z**.
5. Press **Enter** again to return to the privileged prompt.

You can use the `show ip interface s0` command to check the configuration of the serial interface.

One issue relating to the number of IP addresses you have available to configure the routers, hosts, and servers on your network rears its ugly head when you are configuring WAN interfaces. An entire subnet (an entire range of IP addresses) must be wasted to configure the serial interfaces on two routers that are connected by a particular WAN connection.

For example, in the case of our two 2505 routers in Figure 11.1, they are connected by their serial 0 interfaces (using a particular WAN connection and protocol). This connection must be configured as a separate subnet, meaning the serial 0 interface on the Branch Office A router will use one address in the chosen subnet range and the serial 0 interface on the Branch Office B router will use one address from that same subnet range. So, you basically fritter away all the other addresses in that subnet range.

To overcome this obvious waste of IP addresses, you can configure your serial interfaces without IP addresses (they will still route IP packets even though they are designated as *IP unnumbered*). The command used at the configuration prompt for the interface is `ip unnumbered [interface or virtual interface]`. The `interface or virtual interface` parameter is the designation of an actual interface, such as Ethernet 0, or a virtual interface such as loopback 0, that has been configured with an IP address (see Figure 11.5).

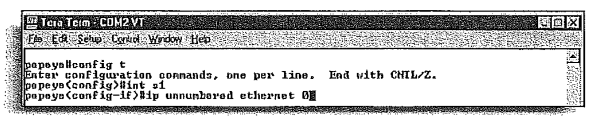


FIGURE 11.5
Serial interfaces can be configured as ip unnumbered, which saves IP addresses for other routers and nodes on your network.

If you use `ip unnumbered` on a serial interface, the serial interface that it connects to via a WAN connection must also be configured as IP unnumbered. The drawbacks of configuring a serial interface as IP unnumbered, is that you cannot Telnet to that serial interface or ping that interface (because it doesn't have its own IP address). Also, if the interface to which you "hooked" the serial port, such as Ethernet 0 (shown in Figure 11.5) goes down, you might not be able to reach the connection that the serial interface is attached to.

Configuring a Routing Protocol

After you have the interfaces on the router configured with the appropriate IP addresses and subnet mask, you can configure a routing protocol. Different Interior Routing Protocols (protocols used for routing on your internal internetwork) are available and your choice of a routing protocol will depend on the size of your internetwork. For example, RIP is fine for small internetworks but is limited to 15 hops (from router to router), making its use on large internetworks a problem. For larger internetworks you may want to use IGRP or OSPF. You will look at the configuration of RIP and the configuration of IGRP in the next two sections of this chapter.

SEE ALSO

➤ *For an overview of IP routing protocols such as RIP and IGRP, see page 93.*

Enabling IP routing

If IP routing has been disabled on the router (it is enabled by default), you will want to enable it before configuring your routing protocol. At the config prompt, type the global command `ip routing`, and then press **Enter**. To exit the Configuration mode press **Ctrl+Z**. If for some reason you want to disable IP routing on a router, you can use the configuration command `no ip routing`.

Configuring RIP

RIP is a distance-vector routing protocol that uses hop count as its metric. RIP summarizes the information in the routing table by IP network numbers (also referred to as major network numbers).

Configuring RIP is very straightforward. You must first select RIP as your routing protocol and then let RIP know the major network number for each interface you have enabled for IP routing. In the sample network that you have been discussing (see Figure 11.1), you are working with only one major network number, 130.10.0.0. So, you only need to specify this network when configuring RIP on our router.

Configuring RIP

1. At the privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. At the config prompt, type `router rip`, and then press **Enter**. This selects RIP as the routing protocol.
3. Type `network [major network number]` at the config prompt. The *major network number* is the network address for a class A, B, or C network that is directly connected to the router. In your case, you are connected to one major network 130.10.0.0. Therefore, the command would be `network 130.10.0.0` (see Figure 11.6). Press **Enter** to continue.
4. Repeat the `network [major network number]` for each IP network that the router is directly connected to. For example, if different Class C networks are connected to several Ethernet interfaces, you must repeat the `network` command for each of the network addresses for these Class C networks.

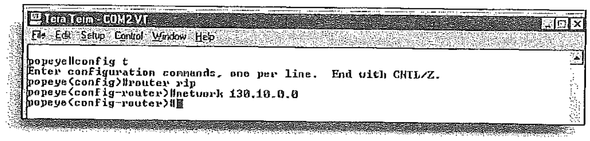


FIGURE 11.6
Router RIP selects RIP as the routing protocol and the network command specifies IP networks connected to the router.

5. When you have finished entering the directly connected networks, press **Ctrl+Z** to end the configuration session.
6. Press **Enter** to return to the Privileged prompt.

After you've configured RIP on your router, you can use the IOS commands that provide a view of RIP routing information such as the routing table and the settings for RIP broadcasts.

To view the RIP routing table, type `show ip route` at the user or privileged prompt and then press **Enter**. Figure 11.7 shows the results of this command on a 2505 router that is connected to another 2505 router via a serial connection. Subnets that are directly connected to the router are marked with a C (interfaces that were configured on that router). Other subnets that are reached by a particular directly connected subnet are marked with an R (these network locations are learned by RIP).

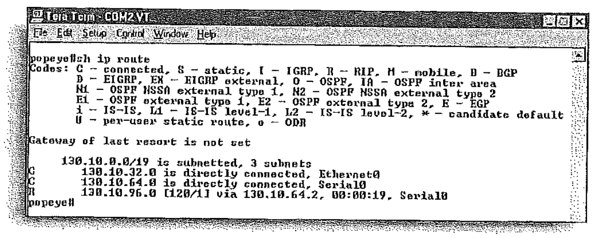


FIGURE 11.7
The `show ip route` command provides a view of the RIP routing table on the router.

You can use the `show ip protocol` command to view the timing information related to RIP. For example, RIP updates are sent every 30 seconds. The hold-down time for RIP is 180 seconds. This means that if a router doesn't receive a RIP update from a connected router, it waits 180 seconds from the last received update and then flags the subnet path as suspect. After 240 seconds, the router will actually remove the path information related to the other router from the routing table because it considers the path no longer usable.

Type `show ip protocol` at the user or privileged prompt and then press **Enter**. Figure 11.8 shows the results of this command.

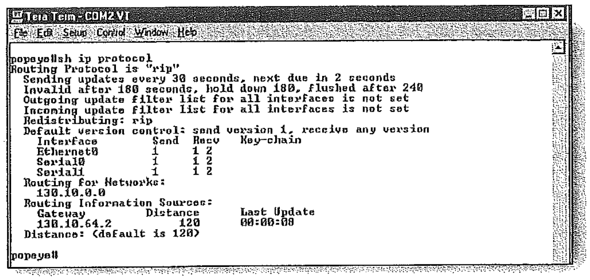


FIGURE 11.8
The `show ip protocol` command provides a view of the RIP timing settings and the networks that are provided routing by RIP.

If you want to view RIP update messages as they are sent and received by a router, you can use the `debug ip rip` command. Type `debug ip rip` at the privileged prompt and then press **Enter**. Figure 11.9 shows the results of this command.

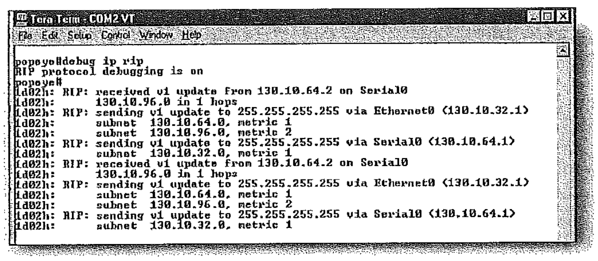


FIGURE 11.9
Use the `debug ip rip` command to view RIP updates on the router.

To turn off RIP debugging, type `no debug ip rip` and press **Enter** (otherwise the update messages will drive you crazy if you are trying to work on the router).

SEE ALSO

➤ *For information on how routers work and using routing protocols to build routing tables, see page 82.*

Configuring IGRP

Because RIP is limited to routes of less than 16 hop counts, intermediate and large internetworks need a routing protocol that can handle the scale of the network. IGRP is a distance vector routing protocol like (RIP) that uses several metrics such as delay, bandwidth, and reliability. IGRP doesn't use hop count as a metric but it can provide routing information for a path of up to 255 hops, which makes it ideal for large internetworks.

Configuring IGRP is similar to configuring RIP. You must enable the IGRP protocol and specify the major IP networks that are directly connected to the router's interfaces. However, because IGRP is used on larger internetworks (such as a complete corporate network), you must specify the autonomous system number for the autonomous system (AS) that the router belongs to. Several different networks (Class A, B, or C) can be part of a particular autonomous system. Autonomous systems are tied together by core routers that run an Exterior Gateway Protocol, such as Border Gateway Protocol (BGP).

Creating autonomous systems

In cases where a company merges with another company or a company's network grows in leaps and bounds, you may want to employ autonomous systems (you have to if you are using IGRP as your routing protocol). Autonomous system numbers can be between 1 and 65,655. You arbitrarily assign them to your different internetworks (but use some kind of numbering system to keep it all straight). The autonomous systems are then tied together by large core routers that run an Exterior Gateway Protocol. See Appendix C, "Selected Cisco Router Specifications," for information on the 7500 series of Cisco that might be used as Core routers.

Configuring IGRP

1. At the privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. At the config prompt, type `router igrp [autonomous system number]`, where the autonomous system number is the AS number assigned to the AS to which your router belongs. For example, `router igrp 10` would enable IGRP routing and specify the AS number 10. After entering the command, press **Enter**.
3. Type `network` *`[major network number]`* at the config prompt. The *`major network number`* is the network address for a Class A, B, or C network that is directly connected to the router. In this case, you are connected to one major network, 130.10.0.0, so the command would be `network 130.10.0.0` (see Figure 11.10). Press **Enter** to continue.
4. Repeat the `network` *`[major network number]`* for each IP network that the router is directly connected to. For example, if different Class C networks are connected to several Ethernet interfaces, you must repeat the `network` command for each of the network addresses for these Class C networks.

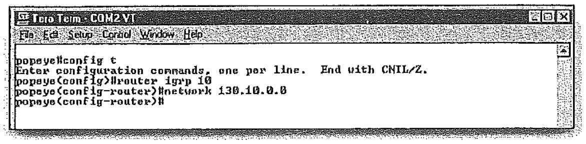


FIGURE 11.10 Router igrp *[AS number]* selects IGRP as the routing protocol and specifies the autonomous system that the router belongs to.

5. When you have entered the directly connected networks, press **Ctrl+Z** to end the configuration session.

You can also use the `show` commands (and variations of these commands related to IGRP) that were discussed in the section on RIP routing. For example, the `show ip route` command now shows the routing table built by IGRP (see Figure 11.11). Network addresses marked with a C are directly connected to the router; addresses marked with an I are those discovered by IGRP.

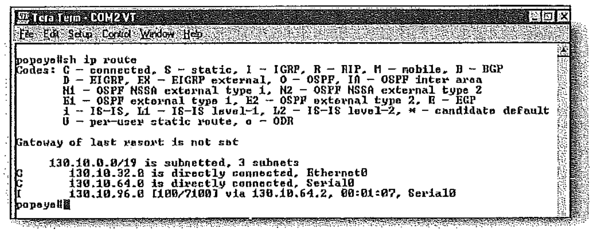


FIGURE 11.11 The `show ip route` command allows you to view the IGRP routing table.

IGRP sends updates every 90 seconds (as opposed to RIP's 30-second interval). Routes not confirmed for 630 seconds are flushed from the router's routing table. You can view this information using the `show ip route` command.

To view a summary of the IGRP routing update messages as they exit and enter the router, use the `debug ip igrp events` command at the Privileged prompt. Figure 11.12 shows the results of this command.

If you want to see information related to the update messages such as the metric used (a number representing a value based on all the IGRP metrics), use the `debug ip igrp transaction` command. Figure 11.13 displays the results of this command.

Turn off all that debugging

To turn off all the debugging you may have enabled on a router, type `no debug all` at the Privileged prompt and then press **Enter**.

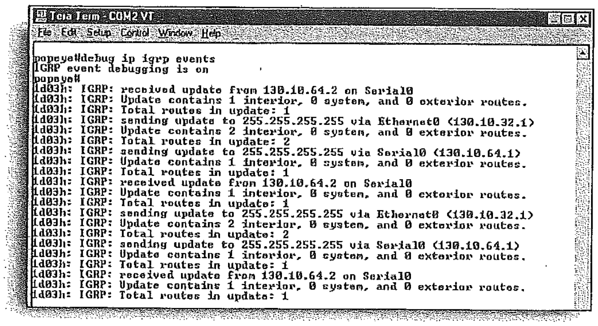


FIGURE 11.12
The `debug ip igrp events` command enables you to view the outgoing and incoming IGRP updates on the router.

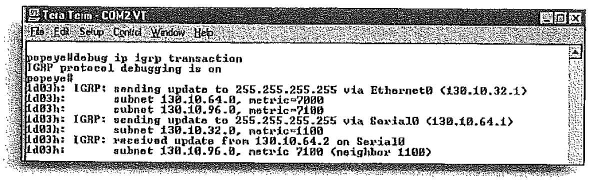


FIGURE 11.13
The `debug ip igrp transaction` command provides information on update messages sent and received and the metric value used.

SEE ALSO

➢ *For background information on IGRP, see page 93.*
➢ *For an overview of Exterior Gateway Protocols, see page 95.*

Dynamic Routing Versus Static Routing

The previous two sections of this chapter enabled the router for dynamic routing. The selected routing protocol (RIP or IGRP) builds a routing table using information received from neighboring routers. You can also configure your routers for *static routing* where you specify the routes in a static routing table. Static routing also requires that you update the routing tables manually.

Static routing doesn't require the use of a routing protocol. You are in charge of the routing tables. However, static routing should probably only be used in cases where the internetwork paths are fairly simple and there is only one route between the network or networks serviced by your router and another router's networks. Static routing tables cannot react to route changes because of lines going down.

Let's keep this simple and use two routers that support networks that have not been subnetted (several different Class C networks). For example, let's say you have two routers connected as shown in Figure 11.14. You want to set up a static route from the router at Branch Office A to the LAN at Branch Office B (Class C network address 194.10.30.0).

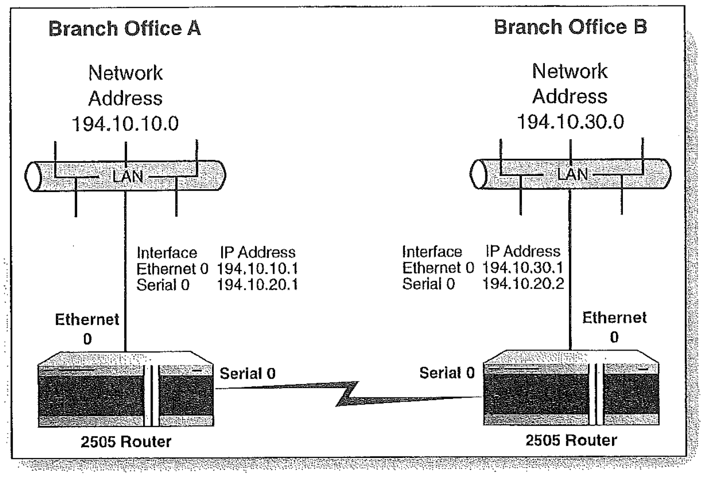


FIGURE 11.14
A small internetwork can be configured for static IP routing.

At the configuration prompt (on the Branch Office A router), you would type the command `ip route 194.10.30.0 255.255.255.0 194.10.20.2`. This tells your router (at Branch Office A) to build a static routing table where network 194.10.30.0 (the LAN at Branch Office B) is reached by the serial connection between the two routers, with the interface on the Branch Office B router configured with the IP address 194.10.20.2. Figure 11.15 shows how this command would look on the router console.

You would have to provide paths for all the routes served by remote routers for your Branch Office A router. And because you have a router at Branch Office B, you would have to use the `ip route` command to configure its static routing table to LANs serviced by other routers (such as the Branch Office A router).

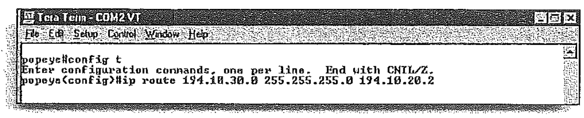


FIGURE 11.15
You configure static routes using the `ip route` command followed by the destination network and the IP address of the router interface on the router that serves the particular network.

As you can see, building your own routing tables statically requires a lot of up-front work. You would also have to update the tables on all the routers involved if any of the routes changed.

Static routing does provide you with complete control over the paths that packets are routed on. However, on large, dynamic internetworks, dynamic routing is probably the way you will want to go when configuring your routers.

Using Telnet

One big plus of configuring IP on your router interfaces is that you can Telnet (connect to) another router using the IP address of one of its interfaces. For example, you have been working with two 2505 routers connected by a serial cable. The router that you are connected to via a serial connection has an IP address of 130.96.1 on its Ethernet 0 port and 130.10.64.2 on its Serial 0 port. You can use either of these IP addresses to gain entry (Telnet) to the other router. After connecting to the router, you must provide the virtual terminal password that was configured on the router.

Using Telnet to connect to another router

1. At the user or privileged prompt, type `telnet [ip address]`, where *`ip address`* is the IP address of one of the interfaces on the other router. To Telnet to the Olive router, directly connected via a serial connection to our Popeye router, type `telnet 130.10.96.1` (the IP address of its Ethernet 0 port), and then press **Enter**.
2. You are connected to the other router and asked to provide the virtual terminal password. Type the virtual terminal password, and then press **Enter**.

You are now logged on to the other router (see Figure 11.16).

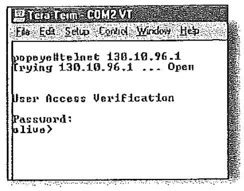


FIGURE 11.16
You can Telnet to a remote router to view its configuration or to configure the router.

If you know the enable password, you can enter the Privileged mode on this router and even change the configuration of the router remotely. When you have finished working on the remote router, type `quit` at the prompt. You are logged off the remote router and returned to the prompt for your local router.

Telnet is a great tool for connecting to remote routers and monitoring or configuring them. It's as if you are sitting at the console computer directly connected to that router.

SEE ALSO

➤ *For information on setting the virtual terminal password when first configuring the router, see page 129.*

chapter

12

Routing Novell IPX

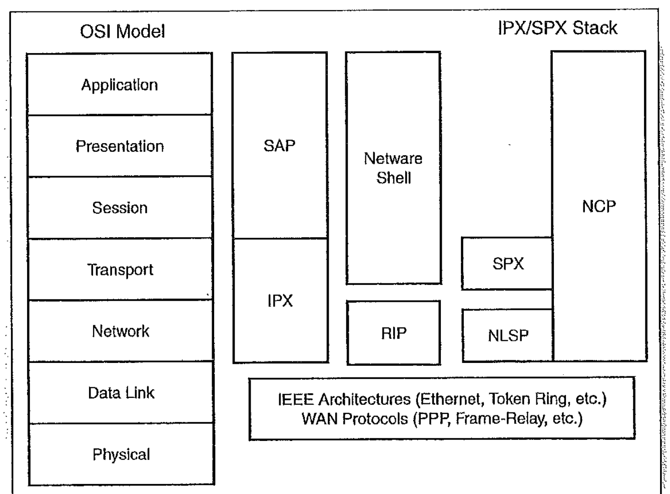
Introducing IPX/SPX

Novell NetWare is a popular network operating system (NOS) that has provided file and print server functionality to LANs since the early 1980s. NetWare has its own proprietary network protocol stack called *IPX/SPX*. IPX is similar to TCP/IP in that the protocols that make up the IPX/SPX stack don't directly map to the layers of the OSI model. IPX/SPX gained a strong foothold in early local area networking because it was strong on performance and did not require the overhead that is needed to run TCP/IP. For many years, NetWare was the leading NOS of choice and can provide client machines with access to LAN and WAN resources.

Novell NetWare is an excellent example of a pure client/server–based NOS. Computers on the network are either clients (who receive services) or servers (who provide services).

IPX/SPX is a routable protocol and so important to our discussion of routing and Cisco routers in particular. Figure 12.1 shows the IPX/SPX stack mapped to the OSI model. The next two sections discuss the protocols in the IPX/SPX stack and how IPX addressing works.

FIGURE 12.1
The IPX/SPX protocol is a routable stack made up of several protocols.



SEE ALSO

- *For a quick review of IPX/SPX in relation to other networking protocols (such as TCP/IP and AppleTalk), see page 48.*
- *For a quick review of the OSI model, see page 34.*

Routing-Related IPX/SPX Protocols

As with TCP/IP, a number of different protocols with different duties make up the IPX/SPX stack. For example, the *NetWare Core Protocol (NCP)* handles network functions at the Application, Presentation and Session layers of the OSI model. The *NetWare VLMs (Virtual Loadable Modules)* establish and maintain network sessions between the client and server. More important to this discussion of routing are the IPX/SPX protocols that are involved in the routing process:

- *SPX (Sequence Packet Exchange)*—A connection-oriented transport protocol that provides the upper-layer protocols with a direct connection between the sending and receiving machines. SPX uses virtual circuits to provide the connection between computers and will display a connection ID in the SPX datagram header (SPX is similar to TCP in the TCP/IP protocol).
- *IPX (Internet Package Exchange Program)*—A connectionless transport protocol, IPX provides the addressing system for the IPX/SPX stack. Operating at the Network and Transport layers of the OSI model, IPX directs the movement of packets on the internetwork using information that it gains from the IPX Routing Information Protocol (RIP).
- *RIP(Routing Information Protocol)*—A routing protocol that uses two metrics, *clock ticks* (1/18 of a second) and *hop count*, to route packets through an IPX internetwork. IPX RIP (like TCP/IP RIP) is a distance vector-routing protocol that builds and maintains routing tables between IPX-enabled routers and NetWare servers.
- *SAP (Service Advertisement Protocol)*—A protocol that advertises the availability of various resources on the NetWare network. NetWare servers broadcast SAP packets every 60 seconds, letting client machines on the network know where file and print

NetWare derived from XNS

In the 1960s, a bunch of geniuses at the Xerox Palo Alto Research Center developed the *XNS (Xerox Network Systems)* network operating system. NetWare is based heavily on this early networking protocol stack. This group of computer scientists and engineers also developed a networked computer that had a graphical user interface and used both a mouse and keyboard as input devices. The technology developed at Palo Alto predates both the IBM PC and the Apple Macintosh. A lot of great ideas came out of this one think tank. So, why doesn't Xerox own the computer world today? Good question.

services can be accessed (each type of service is denoted by a different hexadecimal number in the SAP packets).

- *NLSP (NetWare Link Services Protocol)*—A Novell-developed link-state routing protocol that can be used to replace RIP (and SAP) as the configured routing protocol for IPX routing. The RIP/SAP relationship will be discussed further in the "Configuring IPX Routing" section of this chapter.

As you can see from the discussion of TCP/IP in Chapter 10, "TCP/IP Primer," IPX/SPX is a comparable stack (although it does operate somewhat differently). It has several different protocols that operate at the lower layers of the OSI model (Networking and Data Link) and are involved in the routing process. Before you learn how these protocols interact to make routing of the IPX/SPX packets a reality, you'll learn how IPX/SPX provides an addressing system that defines networks and clients on the network.

Understanding IPX Addressing

IPX addressing uses an 80-bit (10 byte) system (remember, TCP/IP used a 32-bit system), which is comprised of both network and node information, making it a hierarchical addressing system like IP addresses. IPX addresses appear in hexadecimal format and are broken down into two parts. The first part of the address, which can be up to 16 hexadecimal characters in length (this part of the address is 32 bits), is the *IPX network number*. The remaining 12 hexadecimal digits in the address make up the node address (which makes up the remaining 48 bits of the address). Figure 12.2 shows a typical IPX address for a node on a Novell network.

The question then arises as to where the network number comes from and where you get the node address information. I'll discuss the network number first.

When the first NetWare server is brought online in a Novell LAN, a network number is generated during the server software installation. This hexadecimal number becomes the network number for the LAN, no matter how many additional NetWare servers (additional file and print servers) are added to the LAN. So, all client machines (and additional servers) on the LAN will be assigned the same network number (such as 763B20F3, shown in Figure 12.2).

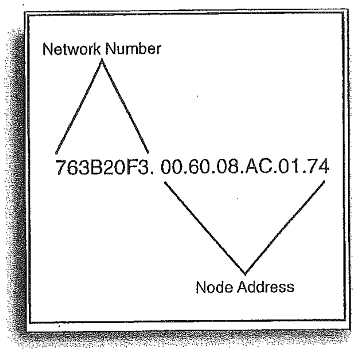


FIGURE 12.2
The IPX addresses consist of a network number and a node address.

So, how do I get the network number?
If you aren't the NetWare LAN administrator, you are, obliviously, going to have to ask the person in charge of the IPX network to give you the network number so that you can correctly configure your router interfaces. If you are the NetWare administrator, load the monitor utility on the server (type `load monitor` at the server prompt, and then press **Enter**) and then open the Network information screens from the Monitor window.

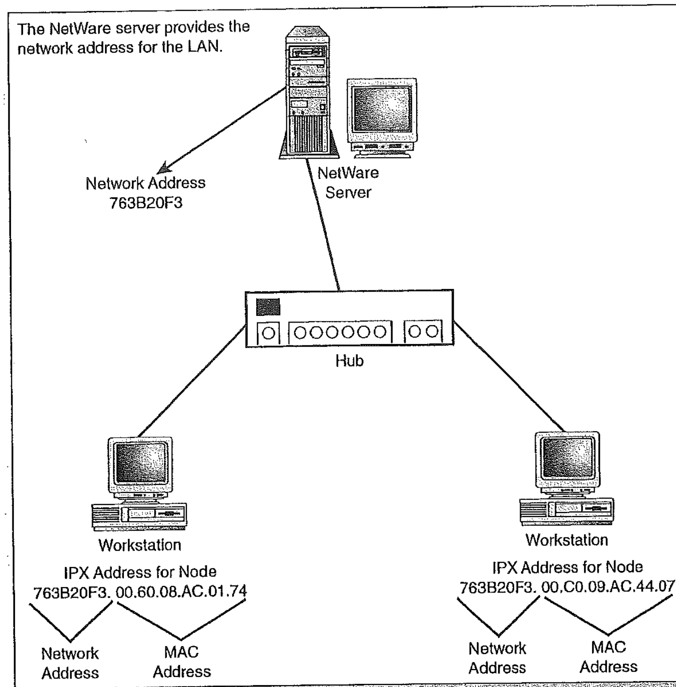
When another new LAN (a separate network entity from the first LAN brought online) is brought into service, its network number will be provided by the first NetWare server brought online on that particular LAN. So, IPX networks are differentiated by their network numbers (whereas IP networks were differentiated by their subnet masks and the subnet bits in the IP addresses). Any routers that play a part in routing packets from a particular LAN will be configured with the network number for that NetWare LAN. This means that the Ethernet 0 interface on the router is connected to a particular NetWare LAN, so it will use that LAN's network number in its interface configuration.

Dealing with the node address for IPX clients is a real no-brainer. It is actually dynamically assigned to the nodes on the network and consists of the MAC address on their network interface card. So, an IPX address is the network number followed by the computer's MAC address. Figure 12.3 shows two nodes and a server on the same NetWare LAN.

SEE ALSO

➢ *For information on MAC hardware addresses, see page 41.*
➢ *For a quick review of network interface cards, see page 13.*

FIGURE 12.3
The IPX addresses for clients on the network will consist of the network number and the client's MAC address.



Understanding SAP

One other aspect of NetWare that I must discuss before I can turn to the configuration aspects of IPX on a router is the part that SAP broadcasts play in IPX networking. Novell servers broadcast SAP announcements every 60 seconds. These broadcasts consist of all the services provided by the server making the SAP announcement and any other services that the server has learned about that are provided by other NetWare servers. The information that a particular server has gathered about other servers and the services they offer is logged in the server's SAP table.

When a particular server broadcasts a SAP advertisement, it is actually broadcasting its entire SAP table and is providing the SAP information to any server (or router) on the network that cares to listen (and all servers do). This means that the SAP information is shared among the servers.

Cisco routers that have interfaces configured for IPX will also build SAP tables and broadcast their SAP information to the networks that the router's interfaces are connected to. Cisco routers don't, however, forward SAP broadcasts from one Novell LAN to another but broadcast their own SAP table (which is a summary of the services offered by each LAN connected to a different router interface). The router provides a summary of all the different LAN SAP tables to each of the different networks. Figure 12.4 shows how the Cisco router would collect the SAP tables from the different networks.

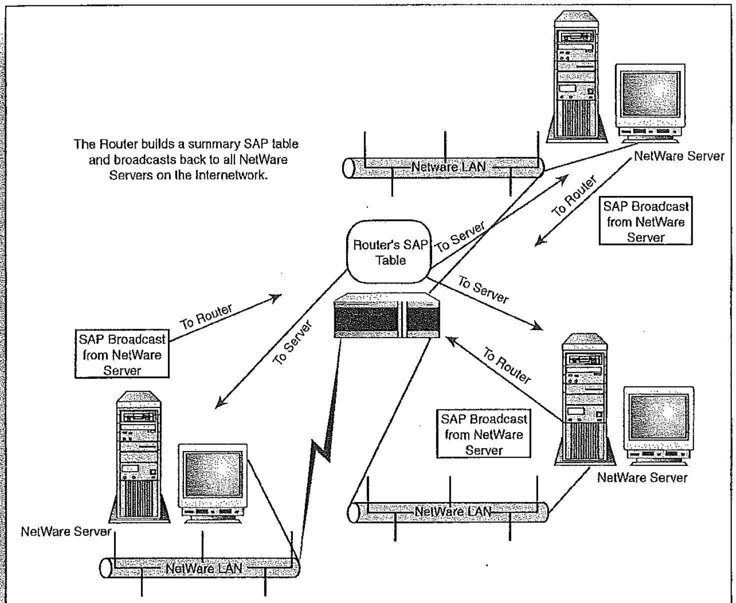


FIGURE 12.4
SAP broadcasts are sent from servers to the router and from the router to the servers.

Client/server communications
When clients on a Novell network want a particular service, they use a broadcast message called a GNS (Get Nearest Server request) to advertise their particular need. A server on the network receiving the GNS will check its SAP table to see where that particular resource (such as a file or printer) is kept. The Server will then send a GNS response to the client, letting it know which server hosts that particular service.

Configuring IPX Routing

Now that you have a feel for how IPX addresses work and the IPX/SPX protocols that are involved in routing, you can configure a router for IPX routing. First, you must start the IPX routing process, and then individual interfaces can be configured.

Enabling IPX Routing

1. At the privileged prompt, type `config t`, and then press **Enter**. This places you in the Global Configuration mode and you are configuring from the console terminal ("t").
2. Type `ipx routing` at the configuration prompt, and then press **Enter**. The configuration command is entered (see Figure 12.5).

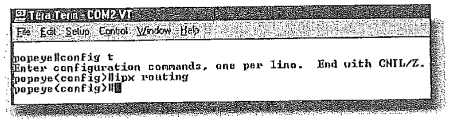


FIGURE 12.5
Enabling IPX routing only takes one global command in the Configuration mode.

3. To complete the process and exit the Configuration mode, press **Ctrl+Z**.
4. You may have to press **Enter** again to return to the privileged mode prompt.

You can easily check and see whether IPX routing has been enabled. Type `show protocol` at the prompt and then press **Enter**. A list of the network protocols enabled for routing will appear (see Figure 12.6). Information related to the protocols is also provided for the interfaces on the router.

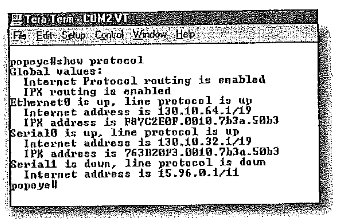


FIGURE 12.6
List the enabled network protocols with the show protocol command.

When you turn on IPX routing using the `ipx routing` command, this also automatically configures IPX RIP as the routing protocol. As mentioned before, RIP uses hop counts and clock ticks as the metric (the RIP used for IP routing used only hop counts). The way in which the two metrics work together is pretty straightforward, if two paths to a particular destination are found (using a router's IPX routing table) that have the same hop count (say, five hops). The more recent of the entries in the routing table, the path with the least number of clock ticks, will be used as the route for the packets. The reverse is also true; when paths have the same tick count, the path with the fewest number of hops is chosen.

Figure 12.7 shows an IPX internetwork where two paths exist between two computers (one sending, one receiving). The paths have the same number of hops (2). If you look at the ticks for the IPX route via Network 2 (a serial connection between router A and one of its neighbors), however, it has only 3 ticks, making it the path that router A will use to forward the packets.

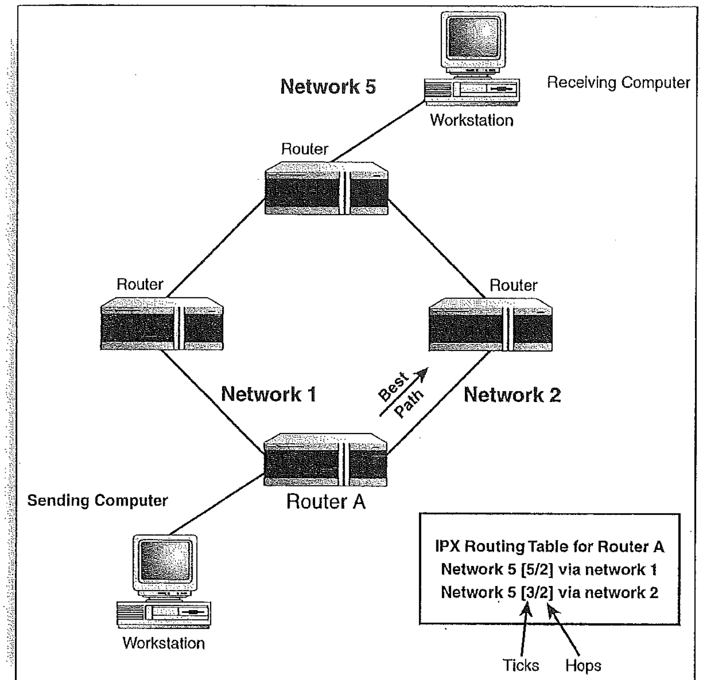


FIGURE 12.7
IPX rip uses hop count and tick count as its metrics.

Configuring Router Interfaces with IPX

After IPX routing has been globally enabled, you can configure the router interfaces that you will route IPX on. The interfaces must be configured for a particular IPX network (dictated by the network

What's the node address?
The format for the command that turns on IPX routing is `ipx routing node`, where *node* is the MAC address of the interface. If you don't enter a node number (MAC address), it is entered automatically for you. Because serial interfaces don't have MAC addresses, they actually "borrow" the MAC address of one of the Ethernet ports on the router for IPX routing. You learn more about this in the section "Configuring Router Interfaces with IPX."

number generated by the first Novell server up and running on that LAN). Because the node numbers are supplied by the MAC addresses of the interfaces, you don't have to worry about that.

It would seem that configuring IPX is easier than dealing with IP routing. But hold on, IPX throws its own curve at you related to the encapsulation type set on LAN interfaces on the router.

LAN Interfaces

Confused yet? Don't despair.

WAN encapsulations, such as HDLC and PPP, were discussed in Chapter 9, "Working with the Cisco IOS." IEEE frame specifications for LANs—such as Ethernet and Token Ring—were discussed in Chapter 2, "The OSI Model and Network Protocols." Encapsulation is basically like placing the data in an envelope. On a LAN, the encapsulation "envelope" (the container for the data) would be an Ethernet frame. On a WAN connection, the data is stuck inside an HDLC envelope until it gets through the WAN connection.

All data on a network is encapsulated in a particular frame type as it moves over the network media as a bit stream. Encapsulation is pretty straightforward with LAN protocols: Ethernet networks use an Ethernet frame, Token Ring networks use a Token Ring frame, FDDI networks use an FDDI frame.

Well unfortunately, NetWare supports more than one frame type for the popular LAN architectures—Ethernet, Token Ring, and FDDI. And if you don't configure your interfaces with the correct frame type or types, they aren't going to talk to nodes on the network or other routers on the internetwork.

NetWare actually supports four different frame types for Ethernet. Because Ethernet networks are so common, Table 12.1 describes each frame type and where you might run into it. The Cisco IOS command (the word you use to set the Ethernet frame type on an interface) is also supplied.

Table 12.1 Ethernet Frame Types

Novell Ethernet Frame Type	Where You Find It	Cisco IOS Command
Ethernet 802.3	Default Frame Type for early versions of NetWare (versions 2–3.11). This is the default frame type chosen when you enable IPX routing on the router.	`novell-ether`
Ethernet 802.2	Default Frame Type for NetWare versions 3.12–5.	`sap`

Novell Ethernet Frame Type	Where You Find It	Cisco IOS Command
Ethernet II	Used in networks running TCP/IP and/or DECnet.	arpa
Ethernet SNAP	Used in networks running TCP/IP and/or AppleTalk.	snap

NetWare supports multiple Token Ring and FDDI frames

Not only does Novell support more than one Ethernet frame type, but it also supports multiple Token Ring and FDDI frames. For Token Ring, it supports standard Token Ring and Token Ring Snap. For FDDI, it supports FDDI SNAP, FDDI 802.2, and FDDI RAW (FDDI frames that don't meet the IEEE specs).

You can specify multiple frame types (encapsulations) on a particular router interface, but each encapsulation must use a separate network number. You are, in effect, using different "virtual" networks to route the different frame types over an interface (when you check the network number on a NetWare server as described in the earlier sidebar, there is actually a different network number provided for each of the different Ethernet frame types).

So, configuring a LAN interface for IPX means that you must supply the network number and the encapsulation type (or types) for the interface. The node address is a given because it is the MAC address of the interface.

Configuring IPX on LAN Interface

1. At the Privileged prompt, type `config t`, and then press **Enter**. This places you in the global configuration mode and you are configuring from the console terminal ("t").
2. To configure an Ethernet port for IPX (such as Ethernet 0), type `interface ethernet 0` at the configuration prompt, and then press **Enter**. The configuration prompt changes to config-if, letting you know that you can now enter the IPX information for the interface.
3. Type `ipx network : ipx network "network number" encapsulation "frame type"`, where *network number* is the NetWare network number provided to you by the NetWare administrator. You must also provide the encapsulation type in this compound command. Suppose you are connecting an Ethernet interface to a Novell network that is running Novell IntraNetWare 4.11. This NOS uses the Ethernet 802.3 frame (the Cisco IOS command is `sap`). Therefore, a complete command would be `ipx network f87c2e0f encapsulation sap` (see Figure 12.8). Press **Enter** to execute the command.

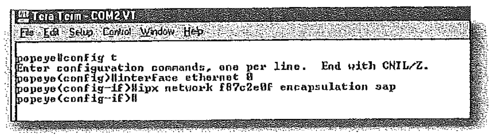


FIGURE 12.8
You must provide the network number and the encapsulation type to configure a LAN interface for IPX.

4. To complete the process and exit the Configuration mode, press **Ctrl+Z**.
5. You might have to press **Enter** again to return to the Privileged mode prompt.

Now you can take a look at the IPX configuration for a particular interface. For example, to check out my Ethernet 0 interface (after configuring it for IPX), I would type `show ipx interface Ethernet 0` and then press **Enter**. Figure 12.9 shows the IPX information for Ethernet 0 on a 2505 router.

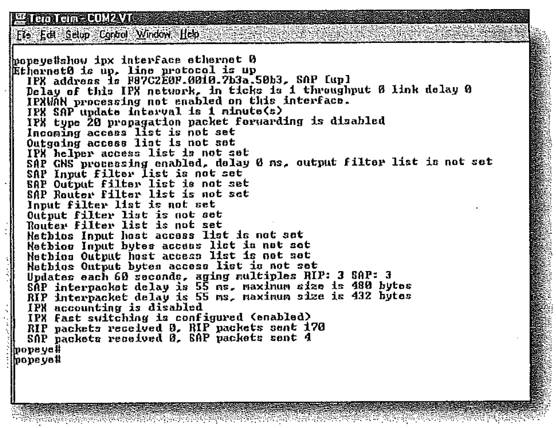


FIGURE 12.9
You can check the network number and encapsulation on a router interface.

You can view all the IPX-enabled interfaces

To view all the interfaces on a router enabled for IPX, type `show ipx interface` and then press **Enter**. All the enabled interfaces will be listed with their network number and encapsulation type.

SEE ALSO

➢ *For an overview of the Cisco IOS and the different IOS modes, see page 142.*

WAN Interfaces

Serial interfaces (using WAN protocols) are configured exactly the same way that you configure a LAN interface for IPX. However, because WAN protocols use their own encapsulation types (they use the frame type supported by the WAN protocol that they are configured for), you don't have to provide an encapsulation type with the IPX network number as you do for LAN interfaces. The encapsulation type for WAN interfaces is set with a separate command where you have to provide a WAN encapsulation method such as PPP or Frame-Relay. The default is HDLC (you will learn how to set different WAN encapsulations like HDLC, Frame-Relay, and PPP in Chapter 15, "Configuring WAN Protocols").

Figure 12.10 shows the configuration parameters for a Serial O interface on a 2505 router. One thing that you must remember when configuring serial interfaces is that two connected serial interfaces (two routers connected to their serial interfaces by a Frame-Relay connection) must inhabit the same IPX network. This isn't unlike IP routing where connected serial interfaces had to be on the same IP subnet.

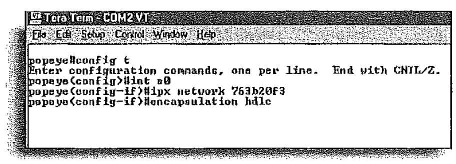


FIGURE 12.10
The IPX configuration of a serial interface.

SEE ALSO

➤ *For a quick review of WAN protocols such as HDLC and PPP, see page 65.*

Monitoring IPX Routing

After you've configured your router or routers to route IPX, you can view the IPX routing tables that are built by the routers. These tables show the networks that the router is directly connected to and other networks that the router has learned about from other routers. You can enter this command in the User or Privileged mode: type `show ipx route`. Then press **Enter**.

Figure 12.11 shows the IPX routing table for a 2505 router connected to another 2505 router via a serial connection. Notice that two networks (763B20F3 and F87C2E0F) are directly connected to the router (denoted with a capital C). Also notice that Network

B86C033F (connected to the Ethernet 0 interface on the other router) is shown in the routing information and can be reached in 7 ticks and 1 hop (7/1).

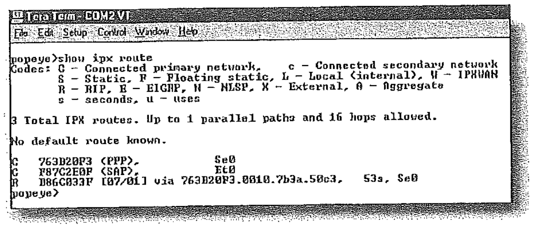


FIGURE 12.11
You can view the IPX routing table for a router.

This network is in the routing table because this router received IPX routing information from the other router that it is connected to. This is how IPX routing tables and routing tables in general, no matter the protocol, are built. Connected routers share information about the internetwork topology.

A couple of other commands that you may find useful when monitoring IPX routing are `show ipx traffic` and `debug ipx routing activity`. The `show ipx traffic` command enables you to view the number and type of IPX packets that have been sent and received by your router. Figure 12.12 shows the syntax for this command and its results.

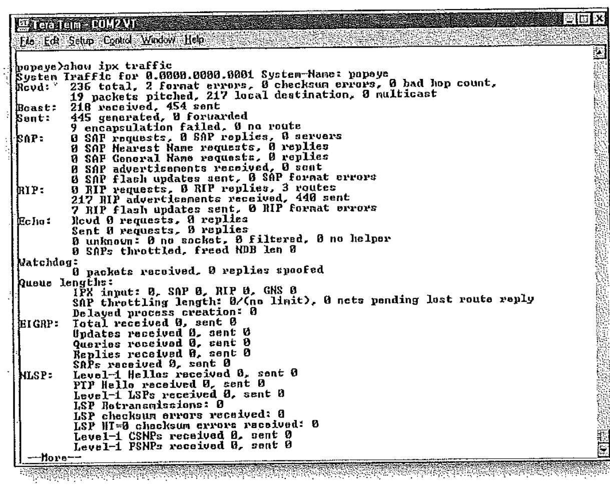


FIGURE 12.12
`show ipx traffic` enables you to view the IPX packets sent and received.

The `debug ipx routing activity` command is a little different in that it doesn't provide a static table of information like the `show ipx route` and `show ipx traffic` commands (those commands are like snapshots of the current status of the router). It actually lets you see the RIP and SAP broadcasts as your router sends and receives them. You must execute this command in the Privileged mode.

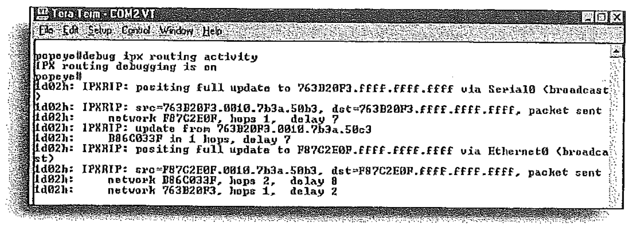


FIGURE 12.13
The debug command enables you to view RIP and SAP broadcast updates as they come in.

After you turn on debugging, you will find that you really can't do a whole lot else on the router because incoming broadcasts keep interrupting your commands. This next command is pretty much worth the price of this book: `no debug ipx routing activity`. This command shuts off the IPX debugging. Never turn it on if you can't remember how to turn it off.

chapter

13

Routing AppleTalk

Understanding AppleTalk

Configuring AppleTalk Routing

Monitoring AppleTalk routing

Understanding AppleTalk

AppleTalk is a routable network protocol stack that provides network connectivity for peer computers (typically Apple Macintosh computers) that want to share files and other network resources such as printers. AppleTalk has its own strategy for network addressing and the grouping of computers into logical workgroups, called *zones*.

Because there always seems to be at least a few Apple computers at every company or institution for multimedia and desktop publishing tasks, it makes sense to be able to route AppleTalk on a Cisco router and allow these computers to share information over an internetwork.

Macintosh computers come equipped with a built-in network interface that can be attached to a hub or other connectivity device using an Apple shielded twisted-pair cable (You have been able to network Macs since they arrived on the scene. The new PowerMacs and G3 computers ship with built-in Ethernet ports). Macintoshes that are integrated into other network architectures can be outfitted with an additional network interface card for that particular architecture (such as an EtherTalk card). AppleTalk supports Ethernet (EtherTalk), Token Ring (TokenTalk), and FDDI (FDDITalk).

Figure 13.1 shows the protocols in the AppleTalk stack that reside at the lower levels of the OSI model. These protocols are used by computers and routers on the internetwork to exchange information such as the location of resources (a server or printer) These protocols are discussed in the following list:

- *DDP (Datagram Delivery Protocol)*—A Network layer protocol that provides a connectionless datagram delivery system similar to UDP in the TCP/IP stack.
- *AARP (AppleTalk Address Resolution Protocol)*—A Network layer protocol that resolves AppleTalk network addresses with hardware addresses. AARP sends broadcasts to all stations on the network to match hardware addresses to logical destination addresses for packets.
- *ZIP (Zone Information Protocol)*—A Network and Transport layer protocol that is used to assign logical network addresses to nodes on the network. This protocol is discussed in more detail in the next section.

- *RTMP (Routing Table Maintenance Protocol)*—A Transport layer protocol that is responsible for establishing and maintaining routing tables on routers that are enabled to route AppleTalk. Routers periodically broadcast routing table information to neighboring routers providing the hops to and the location of AppleTalk networks on the internetwork.
- *NBP (Name Binding Protocol)*—A Transport layer protocol that maps lower layer addresses to AppleTalk names that identify a particular network resource such as a printer server that is accessible over the internetwork.

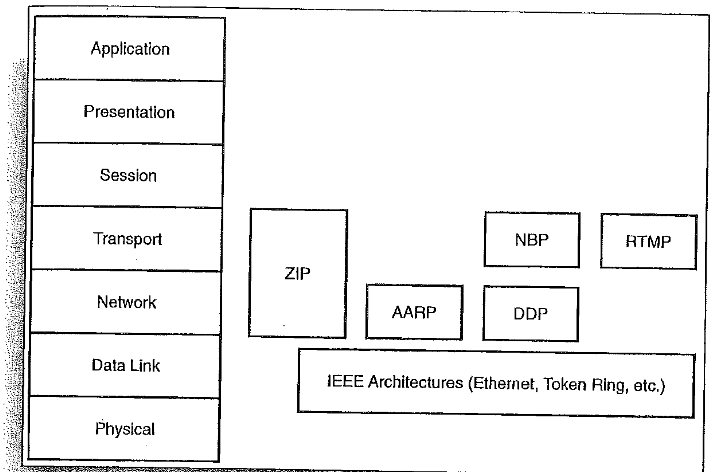


FIGURE 13.1
The routing-associated protocols of the AppleTalk stack mapped to the OSI model.

SEE ALSO

➢ *For general information on AppleTalk in relation to other networking architectures and a look at the AppleTalk protocol stack, see page 49.*

AppleTalk Addressing

AppleTalk uses a 24-bit addressing system that identifies the network segment that the node exists on and the node address itself, which identifies the actual workstation or server.

AppleTalk phase 1 versus AppleTalk phase 2

There have actually been two different phases of AppleTalk: 1 and 2. AppleTalk phase 1 limited the assignment of network numbers to a physical network segment to one network number per physical network. The number of nodes on that network was limited to 127, and the number of servers was limited to 127, making the total number of possible computers 254. AppleTalk phase 2 supplies you with the ability to assign multiple network numbers to the physical network wire and place an unlimited number of nodes and servers on that wire. Phase 2 also allows multiple zones per network. Our discussion of AppleTalk in this chapter will assume the use of AppleTalk phase 2 (which is the appropriate addressing scheme for properly configuring Cisco routers for the routing of AppleTalk).

Dynamic addressing versus static addressing

As already noted, Macintosh computers dynamically generate a network node number on the network. In stark contrast is Novell NetWare (running IPX/SPX) where the node address is assigned statically using the computer's MAC hardware address.

The network address is 16 bits long and the node address portion of the AppleTalk address is 8 bits. Because the number of bits is always fixed for network and node address, you cannot subnet AppleTalk networks as you can with IP addressing. Written in dotted decimal format, the AppleTalk address for particular node would take the format: network.node.

Network addresses are assigned to the various AppleTalk networks by the network administrator and can be a single number designating one network on the network wire or it can be a range of network numbers specifying a number of networks on the same wire. For example, a network address designated as 10-10 means that only one network (network 10) exists on the physical wire that the computers, various hubs, and printers are connected to. A range such as 100-130 would designate multiple networks inhabiting the same network wire. This would be referred to as a *cable range*.

When multiple network numbers inhabit the same AppleTalk network segment this segment is called an *extended segment*. Those with only one network number are called *nonextended*. Each extended network segment can have 253 node numbers associated with each of the network numbers assigned to that particular physical network. Figure 13.2 shows an AppleTalk internetwork with a large LAN made up of extended segments and a LAN that is a nonextended segment. The fact that multiple network addresses can be assigned to the segment (with each network number limited to 253 nodes) makes it possible to put a large number of nodes on any one network segment. Remember that the 8-bit node address limits the number of nodes available, so increasing the number of network addresses available on the network segment increases the number of nodes you can place on it.

AppleTalk node addresses are very easy for the network administrator to deal with because they are dynamically assigned. When a Macintosh comes online with the network, the computer will send out a ZIP broadcast to determine the network number or range of network numbers available on the wire. It will also generate a random node number. The node determines whether the node number is already in use by issuing an *AARP broadcast*.

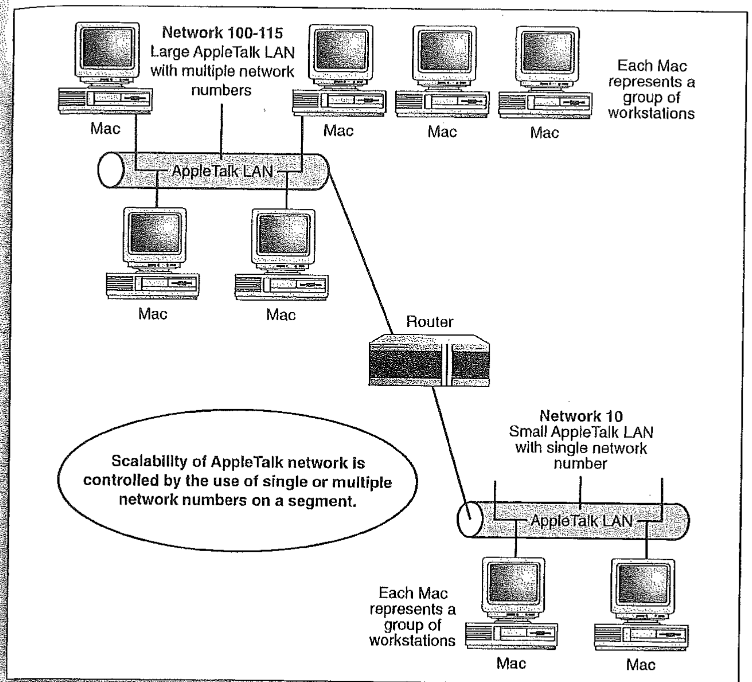


FIGURE 13.2
Extended AppleTalk segments connected by a router.

If the chosen node address on the network number is already taken, the computer will generate another random node address and send out a new AARP broadcast. If the computer finds that all the node numbers are used up on a particular network number, it will choose a new network number and then continue to attempt to take possession of random node addresses on that network (in cases where extended segments have been configured).

After the computer finds a network number and an appropriate node number combination that is available, it will use that address (network.node) as its permanent network address. For example, a computer on network 10 that takes possession of node number 200 would have the permanent address of 10.200.

SEE ALSO
➢ *For information on IP subnetting, see page 180.*

AppleTalk Zones

Another network management tool provided by AppleTalk is the ability to divide the AppleTalk network into zones. *Zones* are logical groupings of users, similar to the concept of workgroups in Microsoft peer-to-peer networking. For example, you may have your desktop publishing staff spread throughout your building; let's say you have Mac users in the Marketing department, some in the Publications department, and so on. You can group these desktop publishers into a logical networking group (known as a *logical zone*) even though they are attached to different segments of the physical AppleTalk network.

Grouping all the desktop publishing staff into the logical zone "desktop" allows these groups to advertise for and access printing and other network services that are spread throughout the building. Routers enabled for AppleTalk will actually build zone tables that can forward broadcast messages from segment to segment on the network, if they are part of the same logical zone.

Zone names are flexible and contain alphanumeric and numeric characters. Marketing1 would be a legal zone name as would destkopA1. Figure 13.3 illustrates the concept of combining AppleTalk LAN segments into the same zone.

Reserved node numbers

AppleTalk does reserve certain node numbers from the pool of 255 numbers—0, 254, and 255. The node number 0 is reserved for temporary use by nodes attempting to determine which network they reside on. Node numbers 254 and 255 are used in broadcast messages to the network, so they cannot be assigne

Configuring AppleTalk Routing

When you enable AppleTalk on your routers and then appropriately configure the router interfaces, the routers will build routing tables that contain network path information much like IP networks. These routing tables allow routers on the internetwork to forward packets on to the appropriate router as the packets move from the sending node to the receiving node.

Before you can configure the router interfaces for AppleTalk routing, you must use a global configuration command to turn AppleTalk routing on.

Learning more about AppleTalk networking

AppleTalk is actually a very sophisticated network protocol stack and as robust and complex as TCP/IP or IPX/SPX. Although you will probably run into AppleTalk less frequently than these other two network protocol stacks, it is still a very viable protocol because Apple computers are common in the desktop publishing and multimedia realms. Because this book is about routers and how they work, the coverage of AppleTalk is limited to broad principles and its addressing system in relation to routing. For more general information on AppleTalk, check out Apple Computer's article library at `http://til.info.apple.com`. Additional documentation on AppleTalk and the Cisco IOS can be found at `www.cisco.com`.

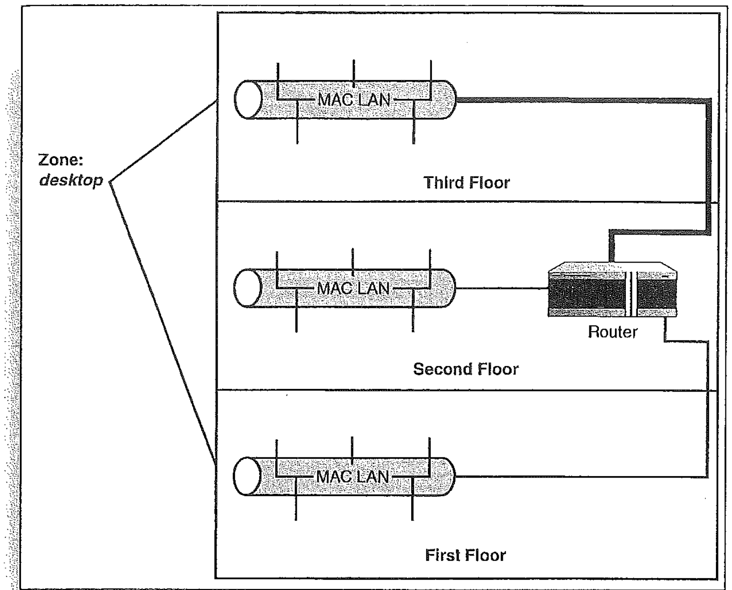


FIGURE 13.3
AppleTalk zones can be used to "join" network segments into one logical workgroup.

Enabling AppleTalk Routing

1. At the Privileged prompt type `config t`, and then press **Enter**.
2. Type `appletalk routing`, and then press **Enter** (see Figure 13.4).
3. To end the configuration session, press **Ctrl+Z**.

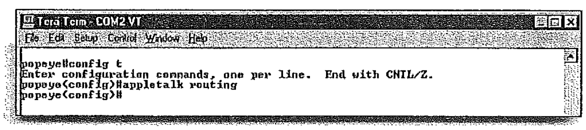


FIGURE 13.4
AppleTalk routing must be enabled on the router before interfaces can be configured.

4. Press **Enter** to return to the Privileged prompt.

When you use the appletalk routing command, RTMP is configured automatically as the AppleTalk routing protocol, so it doesn't have to be configured separately (as RIP and other IP routing protocols did).

Now that AppleTalk routing has been enabled, the interfaces that will be involved in routing AppleTalk packets can be configured. Both the cable range (the range of networks on each segment) and the AppleTalk zones that will be used must be configured on each interface. Figure 13.5 shows two different sites connected using 2505 routers.

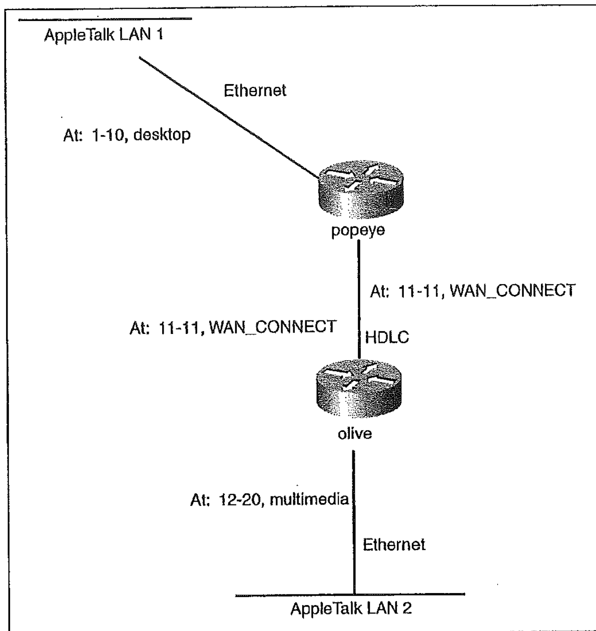


FIGURE 13.5
Two AppleTalk LANS can be connected using two routers that are connected via their serial ports with a WAN protocol and some type of leased connection.

Each LAN uses a cable range (providing a greater number of node addressing possibilities) and the WAN connection uses one network address (which much be configured on the serial port of each connected router). For convenience, the WAN connection is also provided a zone name: WANCONNECT.

Table 13.1 summarizes the configuration information for the AppleTalk network shown in Figure 13.5. We will use this configuration information as examples when we configure the LAN and WAN interfaces for AppleTalk in the next two sections of this chapter.

Table 13.1 AppleTalk Network Configuration Information

Router	Interface	Cable Range	Zone
Popeye	Ethernet 0	1–10	Desktop
	Serial 0	11	WANCONNECT
Olive	Ethernet 0	12–20	Multimedia
	Serial 0	11	WANCONNECT

Configuring LAN Interfaces

Configuring LAN interfaces for AppleTalk is very similar to configuring LAN interfaces for IP or IPX. Network and zone information must be supplied in the Configuration mode for the interface you want to configure.

Configuring a LAN interface for AppleTalk

1. At the privileged prompt type `config t`, and then press **Enter**. You will be placed in the Global Configuration mode.
2. Type `interface ethernet 0` (remember you can abbreviate your commands), and then press **Enter**.
3. At the config-if prompt type `appletalk cable-range 1-10`, and then press **Enter**. (Use the cable range you have determined for your AppleTalk LAN.) This specifies the cable range for the LAN that is connected to the LAN interface on the router.
4. To specify the zone for the interface, type `appletalk zone desktop`. Desktop is the name I am using as a sample LAN zone; you would enter the name of your zone. Then press **Enter** (see Figure 13.6).

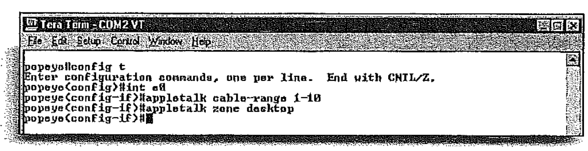


FIGURE 13.6
LAN interfaces must be configured with network and zone information.

5. To end the configuration press **Ctrl+Z**.
6. Press **Enter** to return to the privileged prompt.

This procedure would be repeated for each LAN interface you want to enable to support AppleTalk routing. Remember to provide the correct network range and zone information for each interface. Inadvertently using the same cable range twice would be similar to using the same IP address on two different router interfaces; you won't get the routing that you expect between the networks.

Configuring other LAN types
The example given for configuring AppleTalk on a LAN interface uses an Ethernet interface. AppleTalk also supports Token Ring and FDDI. So if you were configuring a Token Ring interface (the first one on the router) for the routing of AppleTalk, you would supply the network and zone information for the Token Ring 0 interface.

Configuring WAN Interfaces

Configuring WAN interfaces is very straightforward. You must configure the serial ports involved on each router for the appropriate WAN protocol. You must also configure these interfaces with the appropriate network and zone information. Two routers connected via their serial interfaces will have the serial interfaces configured so that they are on the same network and same zone (similar to IP addressing, where both routers must have the connected serial interfaces on the same IP subnet).

Configuring a WAN interface for AppleTalk

1. At the privileged prompt type `config t`, and then press **Enter**. You will be placed in the Global Configuration mode.
2. Type `interface serial 0` (remember you can abbreviate your commands), and then press **Enter**.
3. At the config-if prompt type `appletalk cable-range 11`. Use the network number you have determined for your WAN connection. Then press **Enter**.
4. To specify the zone for the interface, type `appletalk zone wanconnect` (`wanconnect` is used to provide a zone name for the serial connection and also used as a reminder that this is a WAN connection). Then press **Enter** (see Figure 13.7).
5. To end the configuration press **Ctrl+Z**.
6. Press **Enter** to return to the privileged prompt.

SEE ALSO

» *For information on configuring a number of the commonly used WAN protocols on a Cisco router, see page 259.*

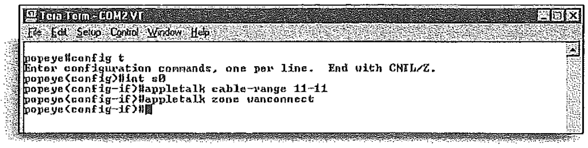


FIGURE 13.7
WAN interfaces must be configured with network and zone information.

Monitoring AppleTalk Routing

After AppleTalk has been enabled on the router and the appropriate router interfaces have been configured, you can view the AppleTalk routing tables on a router and view the configuration of the various interfaces. You can also view statistics related to the AppleTalk traffic on the network including packets sent and received by the router.

To take a look at the routing table for a particular router, type `show appletalk route` at the user or privileged prompt and then press **Enter**. Figure 13.8 shows the routing table for a 2505 router that has its Ethernet 0 interface connected to an AppleTalk LAN and a serial connection to another 2505 router via its Serial 0 interface. The network ranges marked with a C are directly connected to the router. The network range (12–20) marked with an R is another AppleTalk LAN reached via the serial connection to the other router (refer to Figure 13.5 for a diagram showing how these AppleTalk networks are connected).

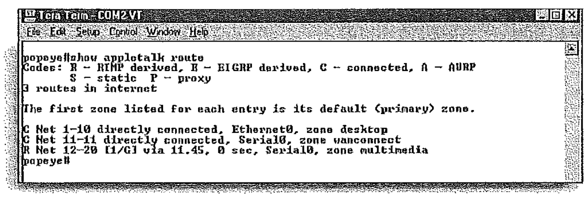


FIGURE 13.8
Use the show appletalk route command to view the AppleTalk routing table on your router.

Several `show related` commands are useful for monitoring the AppleTalk setup on the router. You can view information related to a particular interface or use a broader command that shows AppleTalk configuration information for all enabled interfaces. You can also view AppleTalk zones and their associated network ranges. Table 13.2 provides a summary of some of these commands. These commands can be used at the user or privileged prompt.

show commands provide a lot of information

If you've been going through the chapters in this book in order, you probably noticed that the show commands listed in Table 13.2 are similar to show commands that you used to view information on a router's IP configuration and IPX/SPX configuration information. Learning several of the different show commands, enables you to sit down at any router and quickly get a good picture of how that router has been configured for any network protocol.

Table 13.2 show appletalk Commands

Command	Shows
Show appletalk interface brief	Provides a short summary of all the interfaces on the router and their AppleTalk configurations
Show appletalk interface	Provides more detailed information on the router interfaces and their AppleTalk configurations
Show appletalk interface e0	Enables you to view detailed AppleTalk configuration information for a specified router interface
Show appletalk zone	Provides zone and network information for the zone available on the internetwork.
Show appletalk global	Provides information on the number of networks and zones available on the internetwork and the time interval for ZIP queries and RTMP updates.

Figure 13.9 shows the results of the `show appletalk interface brief` command. Figure 13.10 shows the results of the `show appletalk zone` command and Figure 13.11 provides a view of the results of the `show appletalk global` command.

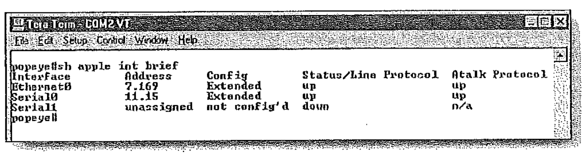


FIGURE 13.9
Use the show appletalk interface brief command to take a look at the interface configurations on the router.

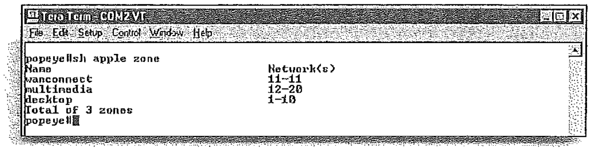


FIGURE 13.10
Use the show appletalk zone command to take a look at the zone and network information on the internetwork.

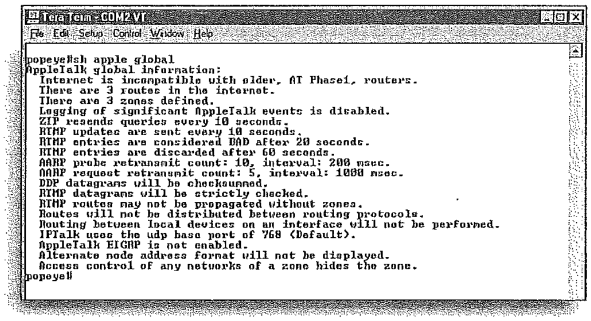


FIGURE 13.11
Use the show appletalk global command to view the overall AppleTalk configuration on the router.

You can also turn on AppleTalk RTMP debugging and view the RTMP routing updates sent and received by the router. Type `debug apple routing` at the privileged prompt and press **Enter**. Figure 13.12 shows the results of this command. To turn off debugging, type `no debug apple routing`, and then press **Enter**. Otherwise, you will find it hard to enter any commands at the prompt.

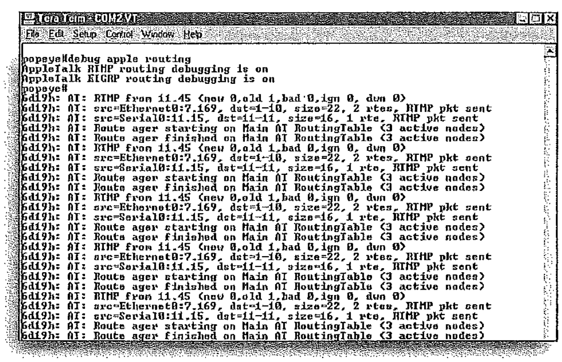


FIGURE 13.12
The results of debug apple routing.

As you can see, AppleTalk provides a routing environment every bit as robust as IP or IPX. And in some ways AppleTalk provides features, such as zones and extended networks, that enable you to easily create complex internetworks of LAN computers at different locations. However, IP still rules the day (and IPX comes in second) so your opportunity to implement AppleTalk routing in the workplace may prove to be very limited.

part

IV

ADVANCED CONFIGURATION AND CONFIGURATION TOOLS

chapter

14

Filtering Router Traffic with Access List

- Understanding Access Lists
- Working with IP Access Lists
- Creating IPX Standard Access Lists
- Creating AppleTalk Standard Access Lists

Understanding Access Lists

So far in this book, you've had a chance to look at how three different LAN protocols (TCP/IP, IPX/SPX, and AppleTalk) are configured on a Cisco router. Interfaces have been configured and connectivity issues relating to creating an internetwork that supports these protocols have been discussed.

But whatyou've basically done is configure your routers so that the doors to your internetwork are hanging wide open. Data packets and broadcast packets have the run of your routers and can enter and leave from any router port they want; you basically have configured a Wild West boomtown without a sheriff. An important part of managing routers and internetwork access is shutting the door on some packets and being a little more selective about what interfaces and routes are available to the data traffic from certain nodes and LANs on your internetwork.

Access lists—a science unto themselves

Working with Access lists gives you a huge amount of control over the data flow on your internetwork. Understanding all the idiosyncrasies of Access lists is a huge task. This chapter gets you started on this subject and covers standard Access lists (you also spend more time working with IP Access lists because IP is the most routed protocol in the world). Extended Access lists can also be built for network protocols such as IP and IPX. For more information, check out `www.cisco.com` or talk to your local Cisco training group (training information is also available on the Cisco Web site). They provide hands-on classes that can help you with a number of advanced subjects related to routers and the Cisco IOS.

This is where an Access list comes in.

The *Access list* is a list of conditions called `permit` and `deny` *statements* that help regulate traffic flow in to and out of a router (and can even control user access to a router via Telnet). A `permit` statement basically means that packets meeting a certain conditional statement won't be filtered out. This means that these packets are "permitted" to continue their journey across the interface. A `deny` statement (by some criterion such as IP address or IPX network address) specifies the packets to be filtered out, or discarded.

Access lists can be used to deny the flow of packets in to a particular router interface or out of a particular router interface. They can also be used to restrict the access capability of certain users and devices to the routers on the internetwork.

How Access Lists Work

As already mentioned, Access lists are a series of conditional statements that can restrict entry of packets from the internetwork to your router based on particular criteria. Each statement in the Access list is read in order, which means that packets coming into a particular router interface are compared to the list criteria from the top to the bottom of the list.

Packets denied are dropped. Packets that are permitted are forwarded as if no Access list existed. If a packet entering the router doesn't match the first statement in the Access list (which can be a `deny` or `permit` statement) the packet is then compared to the next statement in the list.

This process of matching the packet to the `permit` and `deny` statements continues until the packet matches a criteria in the Access list and is either forwarded or dropped. Figure 14.1 illustrates the process of a packet being matched to the `deny` and `permit` statements in an Access list.

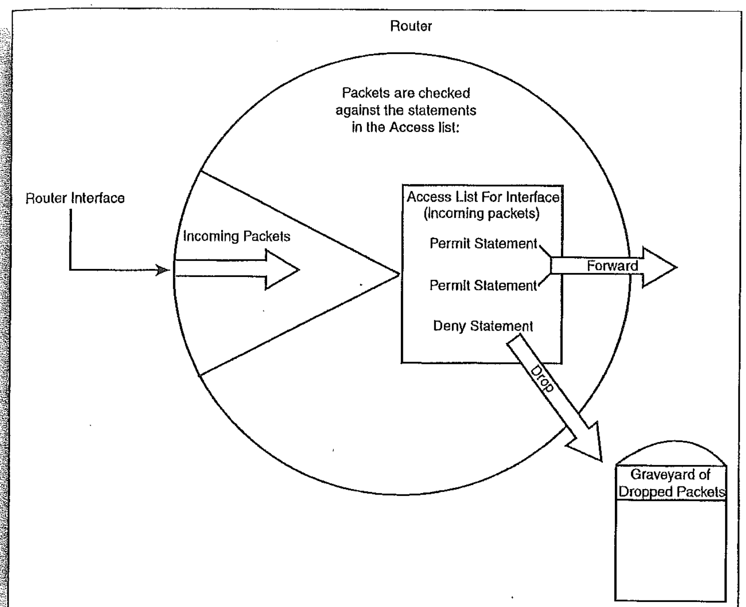


FIGURE 14.1
Packets are either forwarded or dropped based on the statements in the Access list.

A packet that is forwarded from an incoming interface (based on the Access list grouped to that interface) may then face another Access list that is grouped to an outgoing interface on the same router. This means packets can be filtered when received by an interface and then filtered again as it is switched to the departure interface.

For example, you may have a case where you don't want packets entering a router, so you block those packets from entering a particular interface, such as an Ethernet interface that is connected to a LAN. Or you may want to filter the packets as they depart the router. You don't want the packets to leave by a particular serial interface that is connected to another router by a slow WAN connection. You can then assign a filter to this interface, which won't allow packets (addressed in a particular way) to depart from that interface.

Building an Access List

Any interface the router can be grouped to Access lists. But there can only be one Access list associated with the interface for each network protocol that the interface supports. For example, on a router's Ethernet 0 port (which is configured for IP and IPX) an Access list grouped to the interface can exist that filters IP traffic and another Access list can exist that filters IPX traffic. However, you could not have two lists that filter IP traffic grouped to the same interface.

A real plus with Access lists is that you can associate a single Access list to more than one interface on a router. So, for example, the same list could be used by an Ethernet 0 interface and an Ethernet 1 interface on the same router. And you specify whether the Access list is set to filter incoming packets on the interface or outgoing packets. In fact, the same Access list could be grouped to one interface where it filters incoming packets and grouped to another interface on the same router where it filters outgoing packets.

Building an Access list is fairly straightforward; you build the list and then apply it to a particular interface on the router. Be advised, however, that the Access list must contain at least one functioning `permit` statement.

The tricky part of building an Access list is that you have two conditional statements: `deny` and `permit`. You have to determine how you will use these statements to actually limit traffic on the router (without permitting traffic you don't want and restricting traffic you do want).

For example, your strategy might be to use the permit statement to allow access to the router for packets originating on certain LANs on your network (by specifying a separate permit statement that points out each network address that will be permitted). This means that you have several permit statements in the Access list. You can then place a deny statement at the end of the Access list that denies entry to all other networks (which is done in different ways depending on the type of traffic, such as IP packets, that you are filtering).

Or you can use the deny statement to deny entry to certain node or network addresses and then place permit statements near the end of the Access list that allow a number of different networks to move their packets through the interface on your router. Whichever strategy you use, you certainly can't permit a particular network address access to the router through an interface and then deny these same addresses in a later statement. After they hit that permit statement those packets are forwarded, so they are gone even before they are compared to the deny statement.

Creating good Access lists is really a journey in the realm of logic, where you must carefully craft deny and permit statements that forward packets that you want to have routed and drop packets that you don't want routed. And each conditional statement in the Access list must be built so that it doesn't countermand another statement in the list. You certainly don't want the Access list to inadvertently deny the forwarding of packets by your router, when your router is the only path for these packets as they move to their final destination. Let's look at some specific network protocols and how basic Access lists are created for each. This will help shed some light onto the logic of Access lists.

Access lists are a combination of deny and permit statements

You will find that Access lists for interfaces on a router that is part of a fairly good size internetwork will have to weave a filtering web using both deny and permit statements. And after specific nodes and networks have been dealt with in the Access list, a deny all statement (using a wildcard statement based on the network protocol addressing system) is typically placed at the very bottom of the Access list. This denies packets that don't meet any of the conditions you have set in your deny and permit statements.

Working with IP Access Lists

Standard IP Access lists examine the source IP address of packets that are to be filtered on a particular router interface. You use the source IP address as the match criteria for the various deny and permit statements that you place in the Access list.

When designing an Access list that will be used on an interface (such as Ethernet 0 or Serial 1) you must also decide whether the Access list controls the entry of packets on that interface or whether the Access list controls the departure of packets from that interface (which will be forwarded out onto the internetwork). Whether the Access list is for incoming or outgoing packets will have to be specified when the Access list is grouped to the interface. Figure 14.2 shows an IP Access list. I will discuss the commands for creating an Access list in the sections that follow.

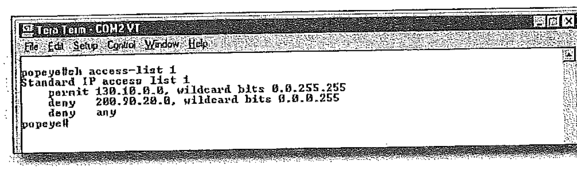


FIGURE 14.2
An IP Access list that permits packets from one network and then denies all others.

Let's take a look at a simple internetwork and use the IP addresses that it provides to create Access lists for some of the routers on the internetwork. Figure 14.3 supplies the information that you will use to create your Access lists.

First, to keep things simple, you will create an Access list for the Serial 0 interface on Router A. You want the data sent from workstation 1A to nodes on the 130.10.0.0 network to be able to use the leased line that connects Router A to Router C as a route. However, you don't want any of the other LANs such as the LAN (200.90.20.0) serviced by router B to use this WAN connection as a possible route (because router B is directly connected to router C). So your list will permit packets from workstation A1 and deny all other packets (from the other LANs).

IP extended access lists

Although our basic discussion of Access lists will examine the use of Standard Access lists for protocols such as IP, you can further fine-tune your network traffic with extended Access lists. In the case of IP, extended Access lists enable you to filter packets based on not only the source IP address, but also the destination address of the packet and particular IP protocols such as UDP and ICMP.

The first step in the process is to create the Access list. The second step in the process is to group the Access list to an interface. However, before you actually create the list, you need to look at one more conceptual item related to IP Access lists—wildcard masks.

SEE ALSO

➤ *For a review of IP addressing, see page 174.*

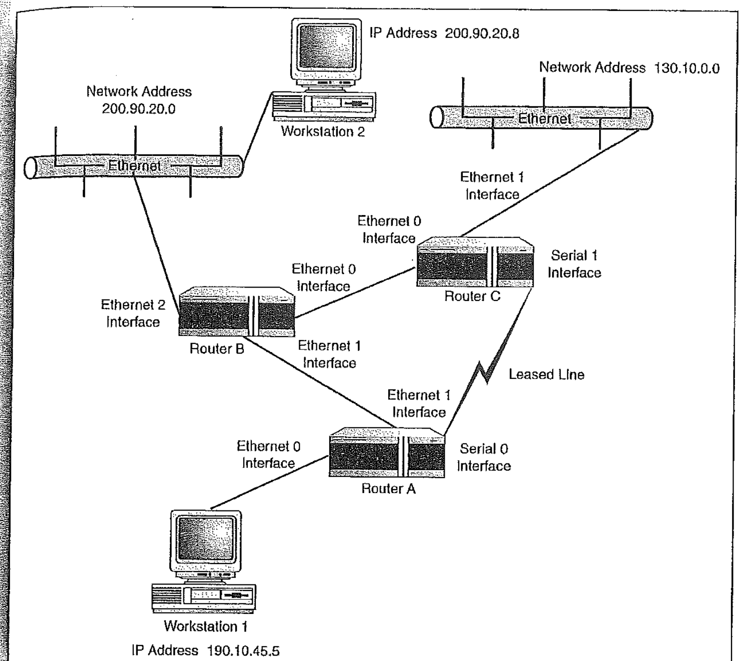


FIGURE 14.3
A simple internetwork crying out for some Access lists.

IP Wildcard Masks

Because the IP addresses used by in basic IP Access lists can be referring to node addresses, subnet addresses, or major network addresses, there must be some mechanism to let the router know which bits in the source IP address of packets that it received should be checked against the IP address provided in the Access list. For example, if the major network address 200.90:20.0 is used in a `deny` or `permit` statement, you want to make sure that the bits in the first three octets are used by the router when it enforces the statement in the Access list on packets that are being processed by one of its interfaces (the interface that the Access list has been grouped to).

You do this with a *wildcard mask*. Bits that you want to have checked in an address must have a wildcard mask value of 0. Bits in the address that you don't want checked are assigned a wildcard mask bit

Wildcard masks are not subnet masks
Don't confuse wildcard masks with subnet masks. Wild card masks are only used in Access lists and their purpose is to let the router know which bits it needs to check in the source IP address of packets to determine whether they should be filtered by the Access list.

value of 1. So, for your major network address 200.90.20.0, where you want all the bits in the first, second, and third octets to be checked by the router, the wildcard mask would be 0.0.0.255 (the binary equivalent of these decimal values would be 00000000 00000000 00000000 11111111).

In the case of a node address (such as 190.10.45.5) where you want all the bits in each octet checked against your entry in the Access list (this would be checked on each packet processed by the interface), you would use a wildcard mask of 0.0.0.0. This means "check all the bits in each octet."

Wildcard mask keywords

In the case of a node address where you want all the address bits checked against the entry in the Access list, you use a wildcard mask of 0.0.0.0. However, you can replace this wildcard mask with the keyword "host," which provides the router with the same mask bits as does the wildcard mask of all zeros. In cases where you want to specify that a `permit` or `deny` statement act on all IP addresses not given in other `deny` or `permit` statements in the Access list, you can use the keyword "any." This is useful if you want a `deny any` statement, which denies all IP addresses except for those placed in `permit` statements in the Access list.

As you can see, when you are working with major network addresses and node addresses, coming up with the wildcard mask is easy. To do this, you would use all zero bits—which equal a decimal value of 0—for octets to be checked, and all 1s or a decimal value of 255 for octets not to be checked. However, when you are dealing with networks that have been subnetted, and you want to permit or deny certain subnets and ignore others (from your range of subnets found on your network), you must construct a mask that tells the router which bits to check in the IP addresses of packets it must process. Let's say that you have subnetted your network (a Class B network) into six subnets as shown in Table 14.1.

Table 14.1 IP Address Ranges for Six Subnets on 130.10.0.0

Subnet #	Subnet Address
1	130.10.32.0
2	130.10.64.0
3	130.10.96.0
4	130.10.128.0
5	130.10.160.0
6	130.10.192.0

You want to create a `deny` statement that will deny packets from subnets 1, 2, and 3 (a subnet range of 130.10.32.0 through 130.10.96.0). This statement would read as `deny 130.10.32.0 0.0.31.255`. The IP address of the first subnet follows the `deny` statement, and the wildcard mask follows the IP address. The big question is how did you come up with the wildcard mask?

For packets to be acted on by the `deny` statement in the Access list, their first octet must match the decimal value 130 so the wildcard mask for that octet in binary will be 00000000—0 in decimal (meaning all the bits in the first octet of the packet must match the binary value of 130 (10000010). And the second octet must match the binary equivalent of 10 (00001010), so again, its wildcard mask will be 00000000 (0 in decimal). So, this means that so far your wildcard mask is 0.0.

Now things become complicated because you are at the third octet where bits have been borrowed for subnetting. Subnet 1 has a third octet value of 32; the binary equivalent of 32 is 00100000. So you have to make sure the third bit is checked (reading the 8 bits from left to right) in packets that are being considered by the router to have the Access list applied to them.

In the second subnet the third octet value is 64 (01000000), so you have to make sure that the second bit in the third octet of the packet is checked. In subnet 3 the subnet value in decimal is 96 (binary value of 01100000), so you need to have the second and third bits checked in a packet to find packets that are in subnet 3.

This means that your wildcard mask from left to right will read 00011111 because you need to check the first bit in the octet (128 to make sure it is off) and you need to check the second and third bit to make sure they are on or off—the 64 and 32 bits. The rest of the bits, 4 through 8, don't need to be checked, so in a wildcard mask these bits are set to 1 (meaning don't check). These bits then have the value 16+8+4+2+1=31. So your wildcard mask for the Access list `deny` statement will read: 0.0.31 for the first three octets in the wildcard mask.

Now you must determine the value for the last octet in the wildcard mask. This octet gives us 8 bits of information relating to node addresses, which you don't want to have checked (the only octet of importance to your router when checking packets against the Access list is octet 3. Octet 4 doesn't have to be checked, so you use a wildcard mask value of 255, which in binary is 11111111). Your complete wildcard mask to filter out (deny) packets from the subnet range 130.10.32.0 through 130.10.96.0 will be: 0.0.31.255.

Remember that wildcard masks aren't subnet masks. The only similarity is that you must convert decimal values to binary values to determine the use of 1s and 0s in the wildcard mask. Now you can create an Access list.

Creating the Access List

So, you will create an Access list that permits packets from workstation A1 (190.10.45.5) to be forwarded out of Serial 0 on Router A but denies packets from all other IP networks. When you create the list you need to assign the list a number from 1 to 99. After the list has been created you will then group it to a particular router interface and at that point make sure to let the router know whether the Access list is filtering packets in or out of the specified interface.

Creating a standard IP Access list

1. At the Privileged prompt type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To create the first line in the Access list type `access-list [`*`list #`*`] permit or deny [`*`ip address`*`] wildcard mask`; where the *`list #`* is a number from 1–99. The statement can only contain deny or permit (not both) and the IP address is the IP address of a particular workstation or network on the internetwork. In your case, you want to block packets from workstation A1 (190.10.45.5), so the command would be `access-list 1 permit 190.10.45.5 0.0.0.0`. Then press **Enter** to continue.
3. To deny all other network packets, type `access-list 1 deny any` (see Figure 14.4).

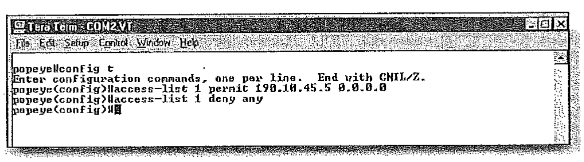


FIGURE 14.4
The Access list is built in the Configuration mode.

4. Press **Ctrl+Z** to end the configuration session.
5. Press **Enter** to return to the privileged prompt.

You can view your Access list using the `show` command. Type `show access-list 1` at the prompt and then press **Enter**. Figure 14.5 shows the results of this command.

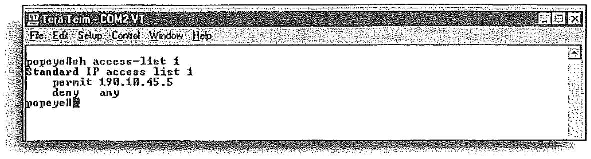


FIGURE 14.5
The results of the `show access-list` command.

Grouping the Access List to an Interface

Now that you have an Access list you can group it to a particular interface on a router. In this case you would want to group this list (which would be created on Router A) to the router's serial 0 interface. You also want the interface to check the packets when they are being prepared to move out of the interface.

Grouping the list to interface serial 0

1. At the privileged prompt type `config t`, and then press **Enter**. You are placed in the global configuration mode.
2. To enter the Configuration mode for serial 0, (or any other interface) type: `interface serial 0`, and then press **Enter**.
3. At the config-if prompt type `ip access-group 1 out` (see Figure 14.6). Then press **Enter**.

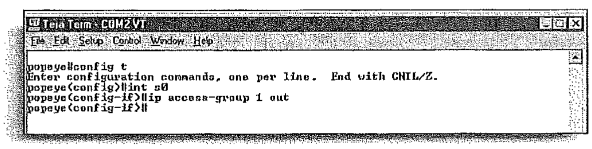


FIGURE 14.6
You must group the Access list to the interface and specify whether the list is used on incoming or outgoing packets.

4. Press **Ctrl+Z** to end the configuration session.
5. Press **Enter** to return to the privileged prompt.

Now that the list has been grouped to the interface, the router can use it to filter packets routing to that interface. You can add additional `deny` and `permit` statements to the Access list. These new statements are added using the same command that you used in the steps

Deleting an Access list

If you find that an Access list isn't working properly (after monitoring traffic on the interface) or you want to start from scratch and build an Access list with the same number, use the command `no access-list 1` (or specify the number of your Access list).

revolving around building your initial Access list. New statements for the list are added at the bottom of the list just above the `deny any` statement (this statement is kind of like a failsafe statement used to block any IP addresses that you missed in your other `deny` statements).

Creating IPX Standard Access Lists

Standard IPX Access lists can deny or permit packets based on their IPX source and destination address. IPX Access lists are numbered from 800–899 (this is the range reserved for IPX Access lists) and are structured similarly to IP Access lists except they use IPX addressing to specify the incoming or outgoing packets that are to be filtered on the router interface.

A typical conditional statement in the Access list would appear as `access list 800 deny [`*`source network address`*`] [`*`destination network address`*`]`. The number `800` (from the IPX Access range of 800-899) tells the router that the Access list is an IPX list. The *`source network address`* would be the IPX network (the network number is provided by the first NetWare server on that network) that serves as the source of the packets. The *`destination network address`* would be the IPX network address of the network that is the intended recipient of the packets.

In IPX Access lists, the value `-1` serves as a wildcard that refers to all IPX networks and is useful in `permit` or `deny all` statements (referring to all networks that aren't listed in more specific `deny` and `permit` statements).

Figure 14.7 shows a simple IPX internetwork. Let's say that you want to build an Access list that will deny packets from network 763B20F3 that are sent to network 02B2F4 via Router C's Ethernet 0 interface.

As with IP Access lists, you must complete two steps. Create the Access list and then group it to the appropriate router interface.

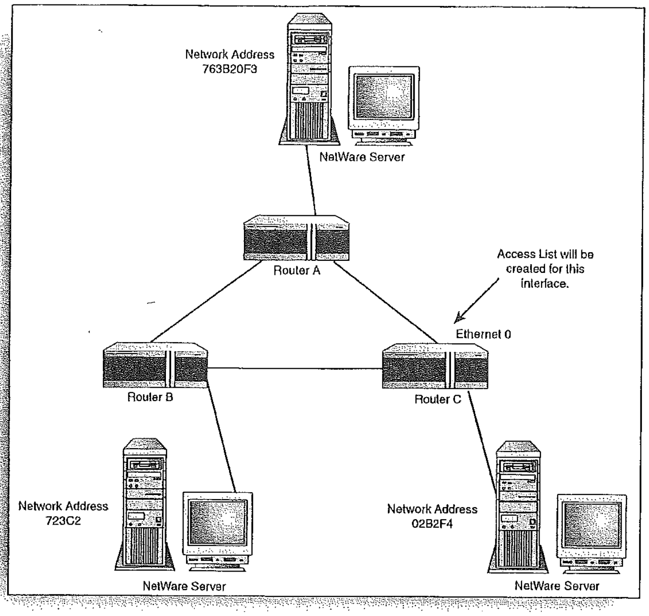


FIGURE 14.7
A NetWare internetwork connected by three routers.

Creating and grouping an IPX Access list

1. At the privileged prompt type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To begin the IPX Access list type `access-list 800` (use any number between 800 and 899 for an IPX list) followed by the source address for the packets, followed by the destination address for the packets. In the case of your sample network (see Figure 14.6) the command would be `access-list 800 deny 763B20F3 02B2F4` (source network address followed by destination address). Then press **Enter**.
3. Add additional `permit` or deny statements as needed. In this case we will add a `permit all` statement for all other IPX networks on our network. Type `access-list 800 permit -1 -1` (permit packets from any network going to any other network). Press **Enter** to continue.

4. To group the list to the Ethernet 0 interface on the router, type `interface Ethernet 0` at the config prompt, and then press **Enter**.
5. At the config-if prompt type `ipx access-group 800 in` (you are filtering packets coming into the interface). Then press **Enter**. Your list is shown in Figure 14.8.

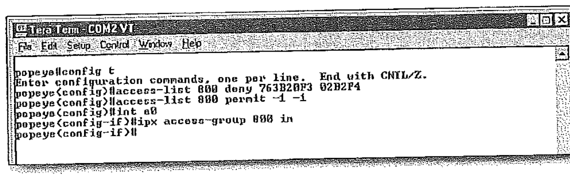


FIGURE 14.8
An IPX Access list under construction that has been grouped to a router's Ethernet 0 interface.

6. Press **Ctrl+Z** to end the configuration session.
7. Press **Enter** to return to the privileged prompt.

You can view your IPX Access list using the `show` command. Type `show access-list 800` (or the number you assigned to your Access list) and press **Enter**.

The Cisco IOS also provides you with the capability to create extended IPX access lists (as it does for IP Access lists) where you can further filter traffic on the network. Extended lists provide you with the capability to filter by network/node source and destination addresses and filter by particular IPX/SPX protocols such as SAP and SPX. Information on Extended list commands can be found on the Cisco IOS Command CD-ROM that is provided with your router.

SEE ALSO

➢ *For a review of IPX addressing, see page 214.*

Creating AppleTalk Standard Access Lists

Access lists can also be built for routers that route AppleTalk traffic. The list numbers reserved for AppleTalk Access lists by the Cisco IOS are 600–699. These Access lists can filter packets based on cable ranges (the network address ranges for a particular physical segment of the AppleTalk internetwork). For example a `permit` statement for an interface may read as: `access-list 600 permit cable-range 100-110`.

AppleTalk Access lists can also be built using AppleTalk zone designations in `permit` or `deny` statements. Using zone designations may serve as a better way to identify parts of your AppleTalk network in `deny` and `permit` statements because Zones can often include more than one cable range. For example, let's say that you have an interface on a router where you want to deny traffic from a particular AppleTalk zone. Figure 14.9 shows a portion of an AppleTalk internetwork.

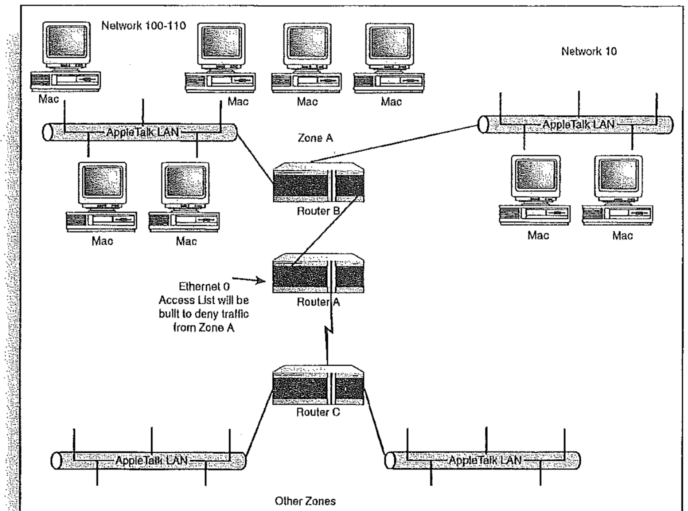


FIGURE 14.9
Access lists can be built to deny or permit traffic from AppleTalk cable ranges and AppleTalk zones.

You want to create an Access list that will deny packets from Zone A (which includes network 100–110 and network 10) on the Ethernet 0 interface of Router A. You also want to make sure that the list allows packets from other zones that might be connected to your network.

AppleTalk uses object names

AppleTalk networks also use object names to refer to servers and other resources on the network. Access list `deny` and `permit` statements can be created using the object keyword followed by the name of the object such as `PrintServer`. Check your router documentation (on the CD-ROM) and `www.cisco.com` for more information on AppleTalk and the Cisco IOS.

Creating and grouping an AppleTalk Access list

1. At the Privileged prompt type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To begin the AppleTalk Access list type `access-list 600` (use any number between 600 and 699 for an AppleTalk list) followed by the zone designation for the packets you want to filter. For the

sample network (see Figure 14.8) the command would be `access-list 600 deny zone ZoneA` (the command word `zone` specifies that you want to set up the statement using a zone name, in this case `ZoneA`). Then press **Enter**.

3. Add additional `permit` or `deny` statements as needed. In your case you will add a `permit all` statement for all the other zones that are on the internetwork. Type `access-list 600 permit additional-zones` (this permits packets from any other zone connected to your internetwork). Press **Enter** to continue.
4. To group the list to the Ethernet 0 interface on the router, type `interface Ethernet 0` at the config prompt, and then press **Enter**.
5. At the config-if prompt type `appletalk access-group 600`, and then press **Enter**. Your list is shown in Figure 14.10.

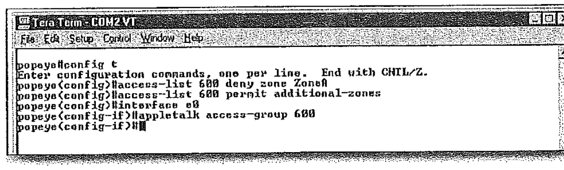


FIGURE 14.10
An AppleTalk Access list under construction that has been grouped to a router's Ethernet 0 interface.

6. Press **Ctrl+Z** to end the configuration session.
7. Press **Enter** to return to the privileged prompt.

You can use the `show access-list 600` (specify the number of your Access list) command to view your AppleTalk list. Dealing with AppleTalk Access lists is actually a little more difficult than IP and IPX lists because of the use of zones and cable ranges to specify network grouping and networks. Considering that you might only deal with small numbers of Macintoshes on a corporate internetwork, the need for filtering might be minimal.

SEE ALSO

➤ *For a review of AppleTalk addressing, see page 229.*

chapter

15

Configuring WAN Protocols

- Understanding Serial and WAN Interfaces
- Configuring High-Level Data Link Control (HDLC)
- Configuring PPP
- Configuring X.25
- Configuring Frame Relay
- Configuring ISDN

Understanding Serial and WAN Interfaces

Most of the discussion so far has been related to connecting LANs to Cisco routers (such as Ethernet LANs using IP, IPX, or AppleTalk as the network protocol), but these routers also enable you to connect routers using a variety of WAN technologies and WAN protocols. The serial interfaces on the router provide the connectivity to the different WAN technologies discussed in Chapter 3, "Wide Area Networking." Routers connecting remotely to other routers using ISDN will typically be outfitted with an ISDN interface.

Is Frame Relay really cost effective?

A local phone company representative quoted me a cost of less than $300 per month for Frame Relay over a T1, which is extremely cheap compared to the cost of this connection a couple of years ago. If you're still using a switched 56K line, check current pricing for Frame Relay over a T1 line. You might be surprised at how affordable Frame Relay has become.

In this chapter, you will look at the Cisco IOS commands that enable you to configure the different WAN protocols on your router or routers. WAN connectivity in general has become much more cost-effective in recent years. Whereas companies once might have used a Switched 56K line connection because of its relative low cost, they now can now use Frame Relay over a T1 line for roughly the same cost.

The type of connection you use will, no doubt, center on cost and line speed. Do your homework before you make the final decision on any WAN connection.

Typically, routers function as Digital Terminal Equipment (DTE) and so a DTE cable would be connected to the serial port on the router and then to a CSU/DSU device (referred to as the Digital Communication Equipment, or DCE) that is then hooked to the line supplied by the phone company. The CSU/DSU device supplies the clock rate for the synchronous transmission.

You can quickly check the encapsulation (the WAN protocol set) for a serial interface using the `show interface serial [`*`interface number`*`]` command. The *`interface number`* would be the serial interface you want to examine. For example, to examine serial 0, the command would be `show interface serial 0` (remember that you can abbreviate your commands). Figure 15.1 shows the results of this command. Note that the interface is currently configured for PPP.

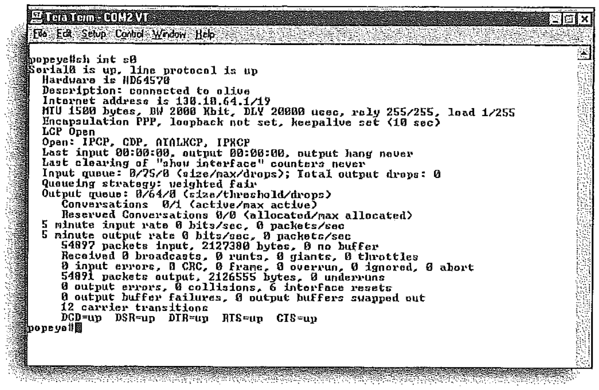


FIGURE 15.1
The show `interface` command can be used to quickly examine the WAN encapsulation for a serial interface on the router.

SEE ALSO

- *For an overview of packet-switching protocols such as X.25 and Frame Relay, see page 62.*
- *For an overview of other WAN protocols such as HDLC and PPP, see page 65.*
- *For an overview of serial interfaces, see page 104.*

Configuring High-Level Data Link Control (HDLC)

HDLC is a point-to-point WAN protocol that serves as the default WAN protocol on Cisco routers. It is already enabled by default. If it isn't enabled on the router, a simple encapsulation command turns HDLC on. One other parameter that you might have to provide when configuring HDLC is bandwidth. This is the throughput of the line that you have leased from the phone company (for example, a 56K line would have a bandwidth of 56). Bandwidth is measured in kilobits/second and is a necessary parameter if you are using IGRP as your routing protocol because IGRP uses bandwidth as one of its metrics.

If HDLC isn't the current WAN protocol for a particular interface, it is quite easy to enable it on a serial interface.

Turning a router into a DCE

You can actually make a router act like a DCE device. In fact, this book was written with two 2505 routers connected with V.35 cables where one router was configured as the DTE and one was configured as the DCE. A V.35 DTE cable was used on the DTE and a V.35 DCD cable was used on the DCE. These two cables where then hooked together (one is female the other male) making the routers think they were connected by a WAN connection. The DCE router had to be configured to provide the clocking that normally is provided by the CSU/DSU device. The `clock rate` (at the config-if prompt for the serial interface being configured) command was used to set the clock-rate on the router. Clock-rate is actually set in bits per second and can range from 1,200 to 8,000,000 depending on the connection. You can check the type of cable attached to your router (DTE or DCE) using the `show controller serial [interface number]` command.

Configuring HDLC on a serial interface

1. At the privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To configure a particular WAN interface, type the name of the interface at the prompt, such as `interface serial 1`. Then press **Enter**. The prompt changes to the config-if mode.
3. Type `encapsulation HDLC`, and then press **Enter**.
4. If you need to set the bandwidth for the interface, type `bandwidth [kilobits/second]`, where the *kilobits/second* is the speed of the line. For instance, for a 56K line you can type `bandwidth 56`, and then press **Enter** to input a bandwidth (see Figure 15.2).

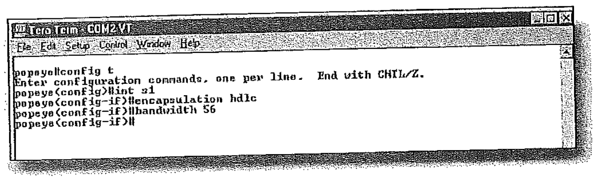


FIGURE 15.2
HDLC is set as the WAN protocol using the encapsulation command. Use the bandwidth command to set the bandwidth if needed.

5. To end the configuration of the interface, press **Ctrl+Z**.
6. Press **Enter** again to return to the privileged prompt.

> **Configuring high-end routers**
>
> Routers that use modular interface slots specify their interfaces in a slightly different way. For example a Cisco 7200 series router denotes a serial interface with the configuration command `interface serial slot/port`. The slot is the modular interface slot on the router and the port is the port number on that slot.

Configuring PPP

Point-to-Point Protocol (PPP) is the TCP/IP stack's point-to-point protocol and can be used for connections between routers using leased lines (in much the same way as HDLC). PPP is an open system protocol and works with IP, IPX, or AppleTalk routing.

Configuring PPP is very straightforward using the encapsulation command. You can also set the bandwidth for the connection as was done for HDLC in the previous section.

Configuring PPP on a serial interface

1. At the privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To configure a particular WAN interface, type the name of the interface at the prompt, such as `interface serial 0`. Then press **Enter**. The prompt changes to the config-if mode.

3. Type `encapsulation PPP`, and then press **Enter**.
4. If you need to set the bandwidth for the interface, type `bandwidth [kilobits/second]`, where the *`kilobits/second`* is the speed of the line. For instance, for a 56K line you can type `bandwidth 56`, and then press **Enter** to input a bandwidth (see Figure 15.3).

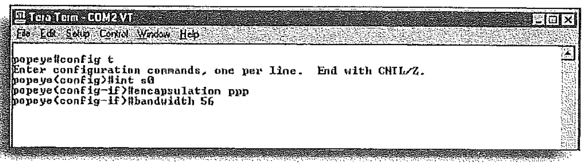


FIGURE 15.3
PPP is set as the WAN protocol for a serial interface using the encapsulation command.

5. To end the configuration of the interface, press **Ctrl+Z**.
6. Press **Enter** again to return to the privileged prompt.

To check your PPP connection to another router, you can use the `ping` command to make sure that both ends of the WAN line are communicating. For example, I have two routers connected or a WAN connection via their serial 0 interfaces. PPP is the encapsulation type. If I know that IP address of the serial interface on the other end of the WAN connection, I can check the line by typing `ping [ip address]`. Figure 15.4 shows the results of the ping command. The destination address was a serial interface on another router with the IP address of 130.10.64.2 (this wouldn't be a possibility if I had configured my serial ports as IP unnumbered).

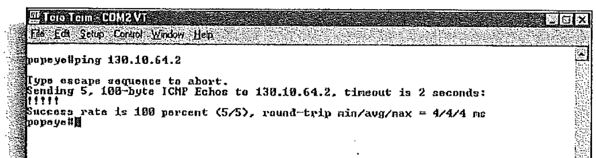


FIGURE 15.4
You can ping a serial interface to check your WAN connection.

Configuring X.25

The X.25 protocol, developed in the 1970s, seems older than the hills now (when compared to Frame-Relay and other more recent and efficient additions to the packet-switching protocol family), but is still employed by companies and institutions for WAN connections. X.25 provides connections between DTEs (such as your router) and DCEs (such as a CSU/DSU).

X.25 uses the *X.121* telephone standards addressing scheme (also known as International Data Numbers) that is comprised of one to 14 decimal digits. This number identifies the local X.121 address for your serial interface and must be configured on the router that is being enabled for X.25.

Depending on the type of X.25 switch your router will connect to, you might also have to set the size of the input and output packets that are moved in to and out of the router over the X.25 connection (the default size is 128 bytes). And again, depending on the type of X.25 switch that serves as your entrance to the X.25 packet-switched cloud, you might also have to set the input and output window size for packets that is used by X.25 flow control (the default window size is 2 packets). All this information should be provided by your connection service provider.

Configuring X.25 on a serial interface

1. At the privileged prompt type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To configure a particular WAN interface, type the name of the interface at the prompt, such as `interface serial 1`. Then press **Enter**. The prompt changes to the config-if mode.
3. Type `encapsulation x25`, and then press **Enter**.
4. To set the X.121 address for the router interface, type `x25 address [data link address]`. The *`data link address`* is the decimal address number (provided by your X.25 provider). For example, you can use the command `x25 address 347650001` (where `347650001` is the X.121 decimal address). Press **Enter** to continue.
5. To set the input packet size, type `x25 ips [bits]`, where *`bits`* is the size of a legal incoming packet. For a packet size of 256, the command would read `x25 ips 256`. Press **Enter** to continue.
6. Output packet size might also have to be set; type `x25 ops [bits]`. To set an outgoing packet size of 256 the command would read `x25 ops 256`. Press **Enter** to continue.

7. To set the window size (based on number of flow control packets) for input to the router, type `x25 win [number of packets]`. You could set the window size to 5 and the command would read `win 5`. Press **Enter**.
8. To set the window out setting, type `x25 wout` followed by the number of packets, such as `x25 wout 5`. Press **Enter** to continue. Figure 15.5 shows the commands entered in steps 1–8 as they appear on the router console.

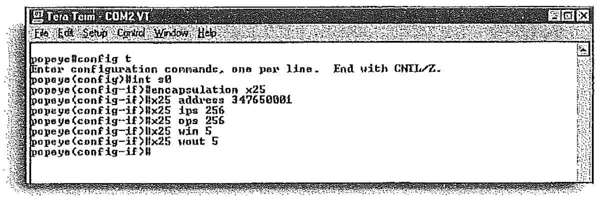


FIGURE 15.5
X.25 encapsulation may require input and output packet sizes and window in and out settings.

9. To end the configuration of the interface, press **Ctrl+Z**.
10. Press **Enter** again to return to the privileged prompt.

You can quickly view your X.25 settings on a serial interface. Type `show interface [serial #]`, where the *serial #* specifies the serial interface that you configured for X.25.

SEE ALSO

➢ *For an overview of X.25, see page 62.*

Configuring Frame Relay

Frame Relay is a packet-switching, Data Link layer protocol that is used to connect DTE (routers) and DCE devices. The DCE devices on Frame Relay networks consist of the carrier-owned switches (see Figure 15.6). The Frame Relay network (a private or public switched telephone network) is typically represented as a cloud.

Frame Relay uses permanent virtual circuits for communication sessions between points on the WAN. These virtual circuits are identified by a *DLCI* (data link connection identifier)—a value provided by the Frame Relay service provider. The DLCI is provided for the connection between the router and the switch (see Figure 15.6) and a DLCI number must be input when configuring Frame Relay on the router.

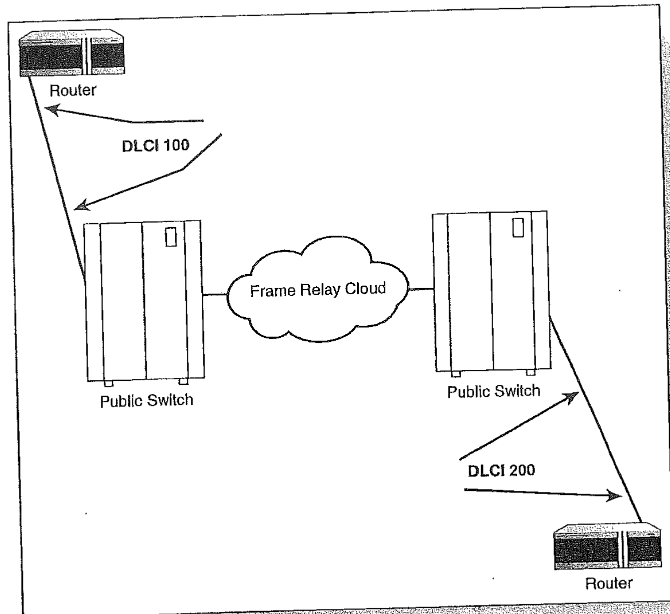


FIGURE 15.6
Frame Relay provides connectivity between routers and public switches.

Another parameter that can be configured for Frame Relay is the *LMI (local Management Interface)*. LMI is the signaling standard used between the router and the Frame Relay switch. Three LMI types are supported by Cisco routers:

- cisco—Cisco, Northern Telecom, DEC, and StrataCom LMI type
- ansi—American National Standards LMI type
- q933a—International Telecommunications standard LMI type

Configuring Frame Relay on the router is similar to configuring the other WAN protocols discussed.

Auto-detect the LMI

Beginning with IOS version 11.2, the router will try to auto-detect the LMI type that is being used on the line between the router and the switch. It will send a request to the Frame Relay switch, which will then respond with the LMI type or types for the line. The router then auto-configures itself using the last LMI type that it receives from the switch (in cases where the switch has sent more than one LMI type response).

Configuring Frame Relay on a serial interface

1. At the Privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To configure a particular WAN interface, type the name of the interface at the prompt, such as **interface serial 0**. Then press **Enter**. The prompt changes to the config-if mode.
3. Type `encapsulation frame`, and then press **Enter**.
4. To set the DLCI for the connection between the router and the Frame Relay switch, type `frame-relay interface-dlci [#]`, where the `#` is the DLCI number provided for the line between the router and the switch. If the DLCI number provided is 100, the command would read `frame-relay interface-dlci 100`. Press **Enter** to continue.
5. The `frame-relay interface-dlci 100` command actually places you at a dlci prompt to configure advanced parameters related to the dlci virtual circuit. To return to the Interface Configuration mode, type `int s0`, and press **Enter**.
6. To configure the LMI (only perform this if you have a version of the IOS older then version 11.2), type `frame-relay lmi-type [LMI type]`, where *`LMI type`* is cisco, ansi, or q933a. To set ansi as the LMI type, the command would read `frame-relay lmi-type ansi`. Press **Enter** after entering the command (see Figure 15.7).

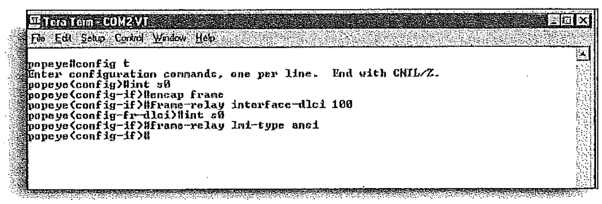


FIGURE 15.7
Frame Relay can be quickly set up on a router serial interface.

7. To end the configuration of the interface, press **Ctrl+Z**.
8. Press **Enter** again to return to the privileged prompt.

After you have configured your router, you can use the `show interface serial [interface number]` command to view the configuration parameters for Frame Relay. Two other commands that are useful for verifying the Frame Relay configuration on your router are `show frame-relay lmi` and `show frame-relay map`.

The `show frame-relay lmi` command provides a listing of invalid messages that have been sent or received by the router and also shows the valid LMI messages that have been sent and received. Figure 15.8 shows the result of this command (you can use the command at the User or Privileged prompt).

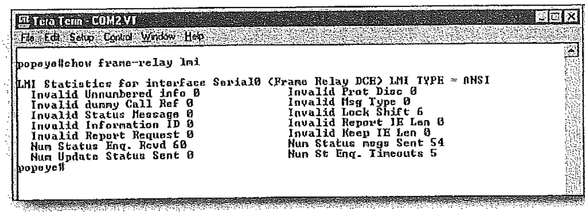


FIGURE 15.8
The `show frame-relay lmi` command provides the status of the LMI type chosen for the router.

The `show frame-relay map` command shows how the DLCI number has been mapped to each of the network protocols that have been configured on the router. For example, Figure 15.9 shows the DLCI 100 mapped to IP, IPX, and AppleTalk.

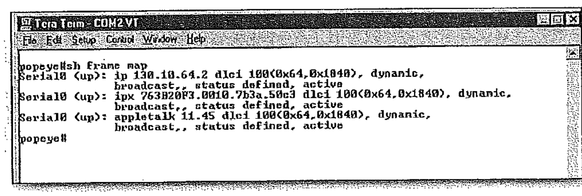


FIGURE 15.9
The `show frame-relay map` command provides information on the mapping of the DLCI number to the network protocols on the router.

One advanced trick to remember is that a single router interface can be configured for multiple DLCI numbers (virtual circuits) using subinterfaces. For example, after configuring interface serial 0, you can specify at the configuration prompt that you want to configure serial interface `serial 0.1`, where the 1 is the first subinterface. You would then configure this subinterface with a particular DCLI number.

Configuring ISDN

ISDN (Integrated Services Digital Network) is a digital service that actually functions over the existing phone lines. It comes in two flavors: Basic Rate ISDN (BRI) and Primary Rate ISDN (PRI).

Typically, if you want to configure ISDN on your router, you want to make sure that you have a router with a built-in ISDN interface. Otherwise, you will have to purchase a *terminal adapter* (also known as an ISDN modem) and connect it to one of the router's serial interfaces.

ISDN is a little different than the other WAN protocols that you've looked at in this chapter. ISDN is the physical conveyance of the data as it moves from a router to the Public Switched Telephone network. It isn't the encapsulation type. You still have to specify an encapsulation type such as PPP or Frame-Relay after you configure the router to use ISDN.

Let's take a look at how you would configure Basic Rate ISDN on a router. Remember that BRI consists of two B channels each providing 64K of bandwidth (which can be combined for a throughput of 128K). Each of these channels must be identified by a *SPID (service profile identifier)*. The SPID number authenticates the channel to the switch that connects the ISDN-enabled route to the phone system. Each channel must have a different SPID number.

Another piece of information that you need to configure ISDN is the *switch type*, which is an identifier code that refers to a particular manufacturer's ISDN switch that you connect to. After you have the SPID numbers and the switch type, all you have to do is provide the encapsulation type for the connection (such as PPP or HDLC).

Configuring BRI ISDN on an ISDN interface

1. At the privileged prompt, type `config t`, and then press **Enter**. You are placed in the Global Configuration mode.
2. To set the switch type for your ISDN connection, type `isdn switch type basic-[switch identifier]`, where the *`switch identifier`* is the manufacturer ID code for the switch type you will connect to. Then press **Enter**.
3. Now you can configure the ISDN interface. Type `int bri [number]`, where the *`number`* is the BRI interface number on the router, such as BRI 0 or BRI 1. Press **Enter**.
4. At the `config-if` prompt enter the encapsulation type (such as `encap ppp`), and then press **Enter**.

Connecting two routers with Frame Relay

If you have the opportunity to connect two routers directly using DTE and DCE V.35 cables (for configuration practice, as you do in the class I teach), you must let the router know that it will serve as a DCE device. During the serial interface configuration, use the command `frame-relay interface-type dce` at the config-if prompt. You will also have to set the clock-rate on the router that you specify as the DCE. To make the router act as a Frame-Relay switch, use the `frame relay switching` command at the global config prompt.

ISDN configuration

ISDN can be configured on a dedicated connection or a dial on demand connection where the router has been configured to dial-up and connect to send and receive data. The router can also be configured to answer incoming calls. Check out the `www.cisco.com` site for more information on configuring ISDN BRI and PRI. Also check out the documentation CD-ROM provided with your Cisco router.

5. To provide the SPID number for the two ISDN B channels at the config-if prompt, type `isdn spid1 [SPID #]`, where the *SPID #* is the telephone number provided by your service provider to reach the particular channel (such as 6125551234). Using this example, the full command would be `isdn spid1 6125551234`. Press **Enter**.
6. To provide the SPID number for the second channel, repeat the `isdn spid2 [SPID #]` command using the SPID number for the second channel. Press **Enter** after typing the command at the config-if prompt.
7. When you have finished entering the outlined information, press **Ctrl+Z** to end the configuration session.

After configuring your ISDN interface you can use the `show int bri[number]` command to view your configuration settings. Make sure that you use the `copy running-config startup-config` command to save the new configuration settings to the router's NVRAM.

SEE ALSO

➢ *For an overview of ISDN, see page 60.*
➢ *For more about NVRAM, see page 113.*

chapter

16

Configuring the Router with Cisco ConfigMaker

- What Is Cisco ConfigMaker?
- Downloading ConfigMaker
- Installing ConfigMaker
- Designing Your Internetwork with ConfigMaker
- Delivering the Configuration to a Router

What Is Cisco ConfigMaker?

ConfigMaker is an incredible, basic router configuration tool that Cisco provides for free. You can download it from the Cisco Web site and it comes with newer versions of the Cisco IOS on a separate CD. You can use ConfigMaker to build your router configuration (you can even build the configurations of all the routers on your internetwork) and then load them onto the routers via your network. If your network isn't up and running yet you can load the router configuration from a PC that is running ConfigMaker and is connected to the router via the Console port.

I've saved the discussion of ConfigMaker until late in the book because, although it is extremely easy to use, it isn't a substitute for an understanding and knowledge of the Cisco IOS commands that are used at the command line on a router console. ConfigMaker is a good way to quickly get a new router up and running, but the fine-tuning of the router configuration will have to be made at the command line. ConfigMaker also doesn't provide any of the router monitoring commands (like `show`, although you can use `ping` from within ConfigMaker).

One hitch in using ConfigMaker to configure a router is that the router must have Cisco IOS 11.2 or newer installed on it (The Cisco IOS was up to version 12.0 at the time of the writing of this book). To check the IOS version on your router use the `show version` command on the router console (at the user or privileged prompt).

If you are using one of the IOS versions that supports ConfigMaker, you're all set. If not, you can still use ConfigMaker to create a network diagram. You can also use it to become more familiar with configuring LAN protocols and their addressing systems on router interfaces.

Downloading ConfigMaker

If you didn't receive Cisco ConfigMaker with an IOS upgrade or with your router, and would still like to use it, you can download it from the Cisco Web site. You can download it even if you don't own a Cisco router, but be advised you cannot use it to configure internetworking devices from other manufacturers. When you do download ConfigMaker from the Cisco Web site, you will have to fill out a registration form.

Connect to the Internet and open your Web browser. In the address box on the Web browser type `http://www.cisco.com/warp/public/734/configmkr`. Then press **Enter**.

On the ConfigMaker Web page that opens, click the **To Download Cisco ConfigMaker, Click Here** link. You will be taken to the registration form page. Fill out the form and then click **Submit**. You will then be provided links to several FTP sites that you can download the ConfigMaker installation file. Select an FTP site and complete the download process.

After the download is complete, you will be ready to install ConfigMaker on your computer.

Installing ConfigMaker

Cisco ConfigMaker runs on Microsoft Windows 95/98–, Windows NT 4.0–, and Windows 2000–based computers. The basic system requirements for running the software are as follows:

- 486 or better (Pentium recommended) computer
- 16MB of RAM
- 20MB of free hard drive space
- SVGA monitor at 800×600 with at least 256 colors
- CD-ROM drive (if installing ConfigMaker from a CD)

As stated earlier, you can install ConfigMaker from a CD-ROM (if you received ConfigMaker with your router or an IOS upgrade) or you can install it from the download version of the ConfigMaker installation program.

For a CD-ROM installation, place the CD in your CD-ROM drive. The installation will start automatically. Follow the prompts to install ConfigMaker to a particular drive and folder on your computer.

If you are installing from the downloaded ConfigMaker installation file, locate the file on your computer using Windows Explorer, and then double-click on the filename. The installation process will begin. Follow the prompts provided to complete the installation.

Now that ConfigMaker is installed on your computer, you can use it to create internetwork diagrams and configure the routers you insert onto the diagram.

Designing Your Internetwork with ConfigMaker

ConfigMaker is really a drawing tool where you create a map or diagram of your internetwork. Icons are available for routers, hubs, LANs, Corporate networks, and a variety of other devices. You basically drag a particular device out onto the network diagram area.

When you drag devices, such as Cisco routers, onto the network diagram, you will be asked to name the device and provide passwords for the device (you will be asked to provide the login password for the router and the Privileged password for the router). In the case of routers, you will also be asked to specify the network protocols (IP, IPX, and AppleTalk) that the router will support.

ConfigMaker handles a number of tasks with easy-to-use Wizards. There is an Address Network Wizard that can be used to address the router interfaces on the various routers in the internetwork and there is a Deliver Configuration Wizard, which walks you through the steps of delivering a router configuration to a router.

The first step in designing your own internetwork with ConfigMaker is to start the software. You can start ConfigMaker from the Windows **Start** menu (click **Start**, point at **Programs**, and then click **Cisco ConfigMaker**) or double-click the **ConfigMaker** icon that was placed on the Windows desktop during the ConfigMaker installation.

Whichever method you use, the ConfigMaker application window will open as shown in Figure 16.1. If this is the first time you've started ConfigMaker you will be asked if you want to view the Getting Started Tutorial; for now let's forgo the tutorial by clicking **No.** This clears the tutorial dialog box from the screen.

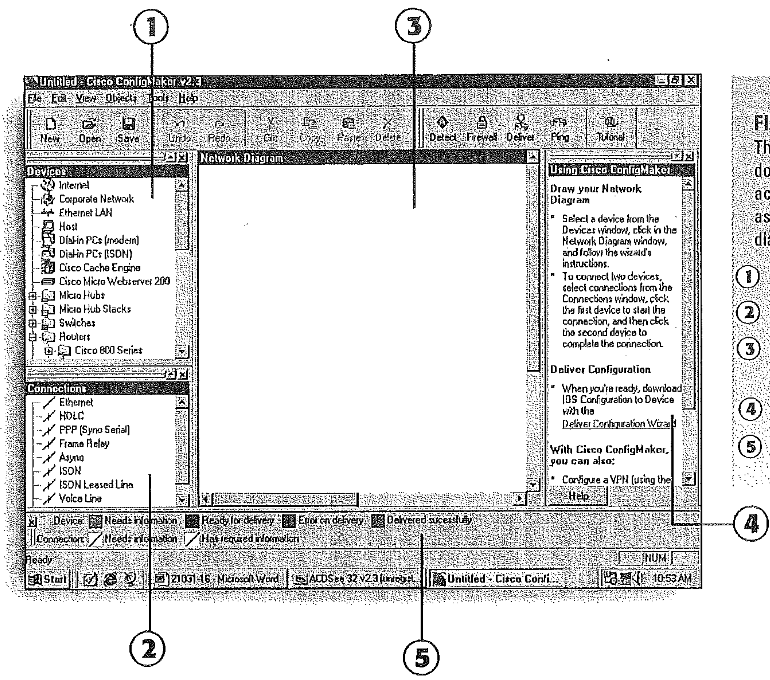


FIGURE 16.1
The ConfigMaker window provides easy access to tools and help as you build your diagrams.

1. Devices window
2. Connection window
3. Network Diagram window
4. Task list
5. Status bar

The ConfigMaker Application window is broken down into several key areas (which are also referred to as windows):

- Devices window—This window provides icons for a number of Cisco devices including routers, hubs, and switches. It also contains the icons for other network devices such as LANs and Corporate Networks.
- Connection window—This window provides the icons for the different types of connections that you can make between the devices that you place in your network diagram. There are LAN connections such as Ethernet and WAN connections such as HDLC and PPP.
- Network Diagram window—This is the space where you build your network diagram using the device icons from the Device window and the various connection icons from the Connection window.

- Task list—This window provides a checklist of all the tasks you must complete to build an internetwork diagram and connect the devices in the diagram. You can hide the Task list to give yourself more room to work in the Diagram window. Click the **View** menu, and then click **Task List** to clear the checkmark and remove the window from the application window (use these same steps to put the window back in the application window).
- Status bar—Provides information on the status of devices when you are loading configurations from ConfigMaker to a device.

Now that you're familiar with the geography of the ConfigMaker window, you can begin to build your internetwork. The first step is to add the devices, such as routers, that will be a part of your internetwork.

Adding Devices

Adding devices to the internetwork diagram is very straightforward. You can add routers (which is of special interest to us, of course) and other devices such as LANs. Let's walk through the steps of adding two devices: a 2505 router and an Ethernet LAN.

Adding routers to the Diagram window

1. First, you will add a 2505 router to the diagram. Scroll down through the Device list until you see the 2505 router folder. Click the **Plus (+)** symbol on the left of the folder to open it. This lists all the routers in the 2500 series family (see Figure 16.2).
2. To add a 2505 router to the diagram, click the **2505** icon and then click in the Diagram window. The Cisco 2505 Router Wizard will appear.
3. In the Device Name box (in the Wizard window), type the name that you want to give to your router (in this case you will use **Popeye**). After typing the name, click **Next**.
4. The next Wizard screen asks you to provide a router password and a Privileged password (see Figure 16.3). Type the passwords you want to use in the appropriate boxes and then click **Next**.

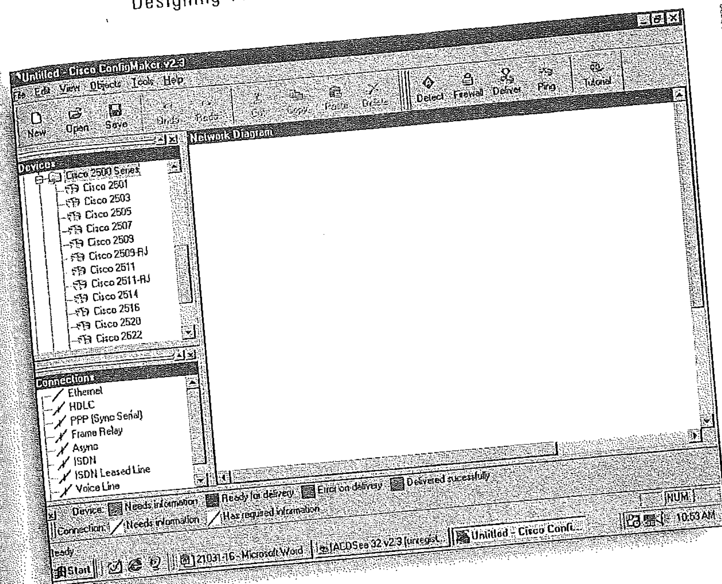


FIGURE 16.2
Router folders are provided that contain the icons for routers that are part of that series.

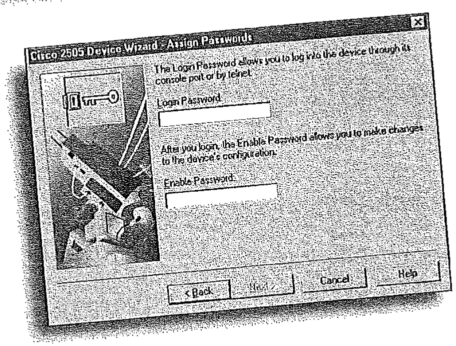


FIGURE 16.3
The Router Wizard asks you to set the login and privileged passwords for the router.

5. The next screen asks you which network protocols you want to enable on this router. IP is the default, but you can also add IPX and AppleTalk (see Figure 16.4). Click the check boxes for the protocols you want to select, and then click **Next**.
6. The last wizard screen lets you know that your router will be added to the diagram. Click **Finish** to end the process.

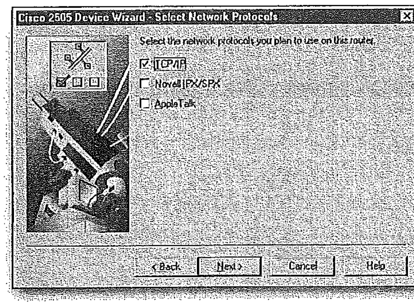


FIGURE 16.4
You can quickly select the protocols that you want enabled on the router.

Your router will appear in the Diagram window. You can change the position of the router in the window by dragging it to a new location. Now that you have a router on the diagram, let's add a LAN that you can connect to the router.

Adding a LAN is very simple. Locate the Ethernet LAN icon in the Devices window. Click the **icon** in the Devices window and then click on the Diagram window where you want to position the LAN icon.

The Ethernet LAN will appear in the Diagram window. Figure 16.5 shows your work so far. You have a router and a LAN in the diagram window. You need to connect them with the appropriate connection type.

> **Getting rid of icons**
> If you select a Device or Connection icon and decide that you've chosen the wrong one, press **Esc** to discard the icon before transferring it to the Diagram window. If you've already placed the device in the window, select the device and press **Delete**.

Connecting LANs to Routers

Connecting LANs to routers is very straightforward. All you have to do is choose the appropriate connection type from the Connection window and then place it between the router and the LAN. At that point you will also have to supply addressing information such as the IP address for the router interface and the subnet mask. If you chose IPX and AppleTalk as supported protocols when you placed the router on the diagram, you will also have to supply addressing information for each of these protocols.

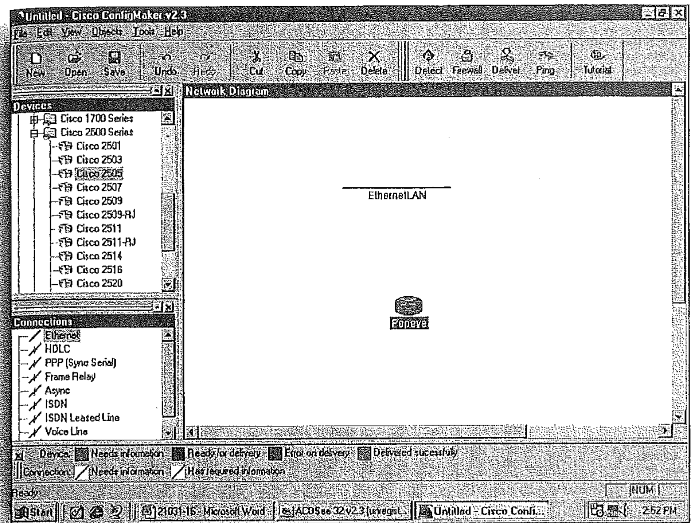


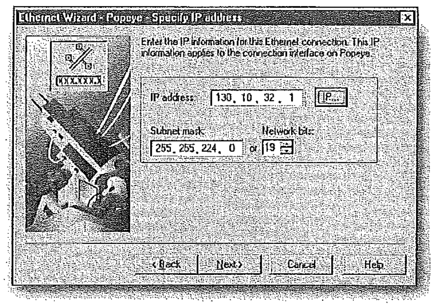
FIGURE 16.5
LANs and routers added to the diagram must be connected.

Connecting a router to a LAN

1. Because you have an Ethernet LAN, the connection between the LAN and the router must be an Ethernet connection. Locate the Ethernet Connection icon in the Connection window. Click on the icon to select it.
2. Click on the router and then click on the Ethernet LAN. This strings the Ethernet connection between the two device icons.
3. As soon as you click on the second icon (the Ethernet LAN), the Ethernet Wizard appears. The wizard helps you set up the connection between an Ethernet LAN and a router Ethernet interface. Click **Next** to begin.
4. You are asked to enter the IP address and subnet mask for the Ethernet interface on Popeye (if you were routing IPX, you would be asked for the IPX network address, for AppleTalk you would be asked the cable range and the zone name). Type the IP address for the router interface in the IP address box (see Figure 16.6).

5. Enter the subnet mask for the interface in the **Subnet Mask** box. You can alternatively enter the number of network bits plus the number of subnet bits used to create your subnets. See Chapter 10, "TCP/IP Primer," to remind you what I'm talking about).

FIGURE 16.6
Enter the IP Address and the Subnet Mask for your router interface.



6. After entering the IP address and the subnet mask, click **Next** to continue.
7. You are told by the last wizard screen that your connection was created successfully. Click **Finish** to close the wizard.

Your connection will appear between the router and the Ethernet LAN. To view the addressing related to the connection (the router interface), click the **View** menu, point at **Attributes**, and then choose either **IP Address**, **IPX Address**, or **AppleTalk Address** (depending on the type of network addressing you are using on the Ethernet LAN and the router). As you know, you can have more than one addressing scheme on the router, so you may want to select more than one option on the **Attributes** submenu.

You can also use the **Attributes** submenu to label the interfaces on the routers that appear in your diagram. Click the **View** menu, point at **Attributes**, and then choose **Port Number**. Figure 16.7 shows the connection that you created between your router and the Ethernet LAN with the router interface labeled with interface number and IP addressing information.

Now that you've seen how to connect a LAN to a router, let's take a look at how you use ConfigMaker to set up serial connections between routers.

The Ethernet Wizard has an IP calculator

If you click the **IP** button on the Ethernet Wizard screen where you enter the IP address and subnet mask for the router interface, you can see the range of addresses that are available in the subnet that you set and the broadcast and network address for the subnet that you are pulling the current IP address from. When you figure out the ranges of IP addresses in your subnet (see Chapter 10), use the IP calculator in the Ethernet Wizard to check your math.

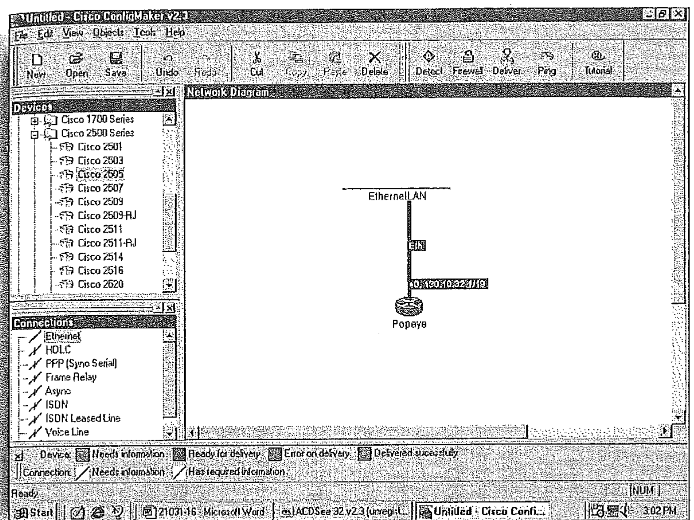


FIGURE 16.7
Use the View menu to turn on some of the view attributes such as addressing, to provide address labeling on the diagram.

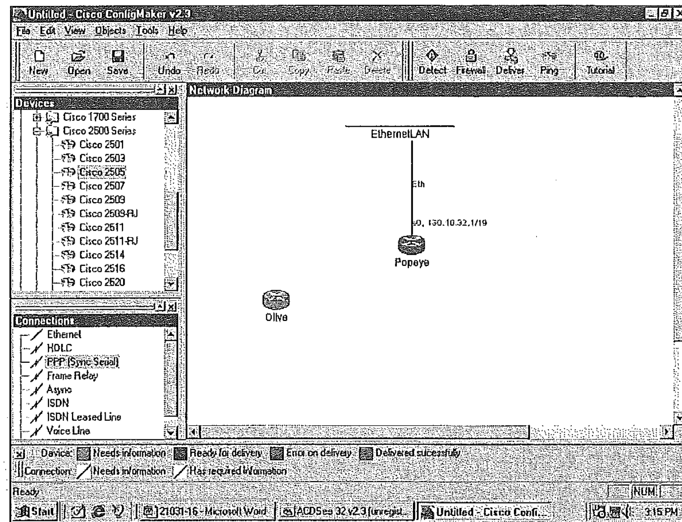
Connecting Routers to Routers

As you already know, routers can be connected using LAN cabling (you can connect two routers in ConfigMaker using the Ethernet connection) or connected to each other remotely using serial connections and a particular WAN protocol such as PPP or Frame-Relay. ConfigMaker makes it very easy for you to create serial connections between the routers on your diagrams. First, you will add another router (it doesn't matter what kind, you may want to explore some of the high end routers in ConfigMaker, even if your company doesn't use them). I've placed another 2505 router on my diagram (see Figure 16.8) and will connect it to the router that is currently in the diagram (Popeye).

Connecting a router to a router with a WAN protocol

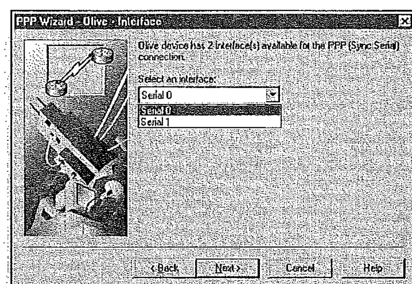
1. With the two routers visible in the Diagram window, click the Wan Protocol connection type (such as PPP) in the Connection window.
2. Click the first router and then click the second router to specify where you want to create the connection.

FIGURE 16.8
Place two routers on the diagram and then you can connect them with a particular WAN technology.



3. As soon as you click on the second router icon, the Wizard for the WAN protocol that you selected (such as PPP or HDLC) will open. In the figures shown in subsequent steps you will see that I chose **PPP**.
4. To begin the connection process click **Next**.
5. On the next screen you are asked to select a serial interface (such as Serial 0) to configure for the WAN connection. Use the drop-down arrow on the Wizard screen to select the serial interface you want to use (see Figure 16.9). Then click **Next** to continue.

FIGURE 16.9
Select the serial port you want to configure for the WAN connection to the other router.



6. On the next screen you are asked to enter the addressing information for the Serial port that you chose (see Figure 16.10). In this case (because I set up the router's to route IP only, you must provide the IP address and subnet mask for the serial interface on Olive. Enter the IP address and Subnet Mask and then click **Next** to continue.

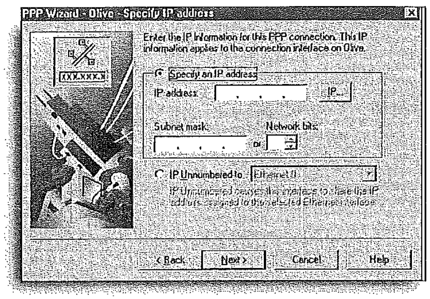


FIGURE 16.10
Provide the addressing information for the selected serial interface, such as the IP address and subnet mask.

7. On the next screen you are asked to select the serial interface on the other router (in this case Popeye) After using the drop-down arrow to select a serial interface, click **Next**.
8. Supply the addressing information (such as IP address and subnet mask) as you did for the other router in step 6. Click **Next** to continue.
9. The next screen asks you if you want to create a backup connection for this WAN connection. In this case, you will go with **No Backup** (the default). Click **Next**.
10. On the last screen you are told that you have successfully created a WAN connection. Click **Finish**.

The connection will be created in the Diagram window (see Figure 16.11). If you have the View Addressing attribute turned on (using the **View** menu), you can see the addressing information for the serial interface on each of the created routers.

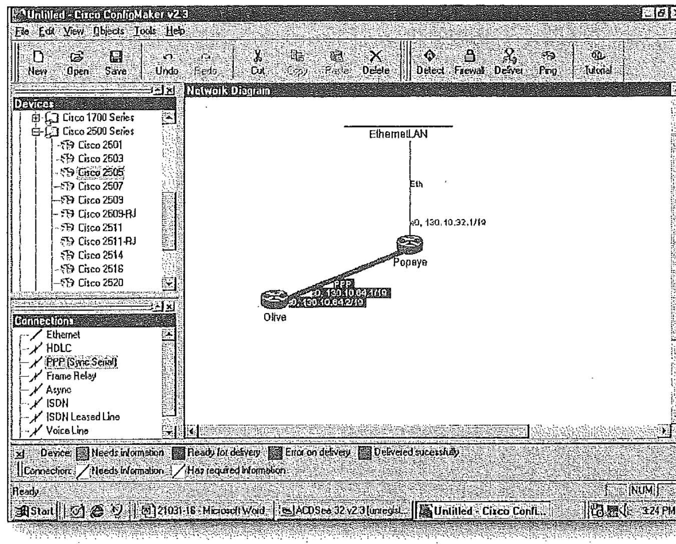


FIGURE 16.11
The new WAN connection will appear between the routers in the Diagram window.

Delivering the Configuration to a Router

You can use ConfigMaker to build an entire internetwork diagram. You can connect LANs and routers, hosts and routers, and connect routers to routers. All the devices that you could possibly need and the various connection types are available in the Device and Connection windows respectively. After you build your internetwork, you can actually use the configuration settings that you provided for your router (or routers) interfaces directly to the router.

You can download a configuration to a router or routers using a PC that is running ConfigMaker and is connected to the same network that the routers are connected to. You must have configured the PC and routers with IP addresses, however, before ConfigMaker can send the configuration over the network. This requires that you "preconfigure" the router using the router console.

An easier method of quickly delivering a configuration to a router that contains no configuration what-so-ever, is to download the configuration from a PC running ConfigMaker that is connected to the router using the console port and console roll-over cable. You would connect the PC to the router as you would connect a PC console.

One thing that you should check before you try to deliver the configuration (as shown in the steps that follow), is that ConfigMaker will deliver the configuration port using the correct serial port on your computer. The default setting is COM port 1.

If you need to change the COM port setting, click the **View** menu, and then select **Options**. In the Options dialog box that appears use the COM port drop-down arrow to select he appropriate COM port and then click **OK**.

Delivering a router configuration using the Console port

1. With the internetwork diagram open in ConfigMaker that contains the router configuration that you want to deliver, select the appropriate router icon (see Figure 16.12). I selected the Popeye configuration).

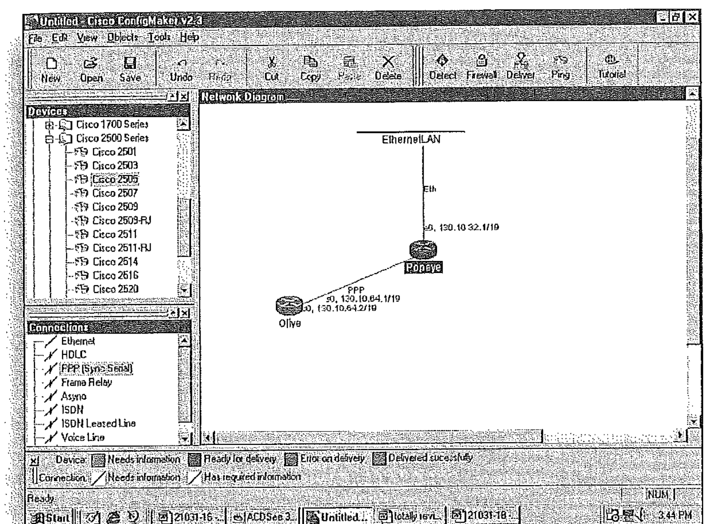


FIGURE 16.12
Select the router icon that will supply the configuration you will deliver to the connected router.

2. Click the **Deliver** button on the ConfigMaker toolbar. The Deliver Configuration Wizard will open, listing the router that you selected in your diagram.

3. Click **Next** to continue. The next screen tells you to make sure that no other programs are using the COM port that will be used to deliver the configuration (if this PC also serves as the router's console, make sure that your terminal emulation software isn't running).
4. When you are ready to deliver the configuration, click **Next**.

That status of the configuration delivery will be displayed on the Wizard screen (see Figure 16.13). The current configuration on the router (if any) will be erased and then the router will be rebooted. The new configuration will then be loaded into the router's NVRAM.

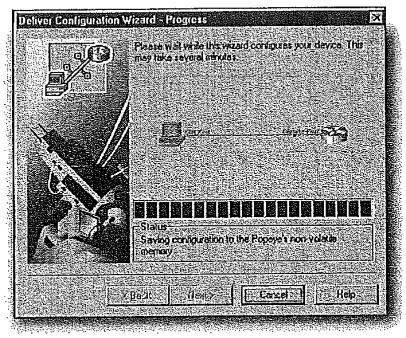


FIGURE 16.13
You can watch the progress of the configuration delivery.

A final wizard screen will appear providing the router name, the delivery method (console) and the time and date that the delivery was completed. Click **Finish** to close the wizard box.

Your router is now configured with the configuration that was downloaded. Use your terminal emulation software and check out the delivered configuration using the `show startup-config` command. You should see the same settings that you placed in the configuration for the router that appeared in your ConfigMaker diagram.

When you have finished working on a particular internetwork diagram, click the **Save** button on the ConfigMaker toolbar. In the Save As dialog box type a name for the internetwork diagram and use the **Save In** drop-down box to specify a drive and folder for the file. Click **Save** in the dialog box to save your diagram.