

Configuring Novell IPX

This chapter describes how to configure Novell Internet Packet Exchange (IPX) and provides configuration examples. For a complete description of the commands mentioned in this chapter, refer to the “Novell IPX Commands” chapter in the *Network Protocols Command Reference, Part 2*.

Note One or more of the commands that previously appeared this chapter have been replaced by new commands. See the *Configuration Fundamentals Command Reference* for command information. The old commands continue to perform their normal function in the current release, but support for them will cease in future releases.

IPX Addresses

An IPX network address consists of a network number and a node number expressed in the format *network.node*.

The network number identifies a physical network. It is a 4-byte (32-bit) quantity that must be unique throughout the entire IPX internetwork. The network number is expressed as eight hexadecimal digits. The Cisco IOS software does not require that you enter all eight digits; you can omit leading zeros.

The node number identifies a node on the network. It is a 48-bit quantity, represented by dotted triplets of 4-digit hexadecimal numbers.

The following is an example of an IPX network address:

```
4a.0000.0c00.23fe
```

In this example, the network number is 4a (more specifically, it is 0000004a), and the node number is 0000.0c00.23fe. All digits in the address are hexadecimal.

IPX Configuration Task List

To configure IPX routing, complete the tasks in the following sections. At a minimum, you must enable IPX routing. The remaining tasks are optional.

- Enable IPX Routing
- Configure NLSP
- Configure IPX Enhanced IGRP
- Control Access to IPX Networks

- Tune IPX Network Performance
- Configure IPX Accounting
- Shut Down an IPX Network
- Configure IPX and SPX over WANs
- Configure Next Hop Resolution Protocol (NHRP)
- Monitor and Maintain the IPX Network

See the “Novell IPX Configuration Examples” section at the end of this chapter for configuration examples.

Enable IPX Routing

To enable IPX routing, you must perform the tasks described in the following sections:

- Enable IPX Routing
- Enable Concurrent Routing and Bridging IRB
- Configure Integrated Routing and Bridging (IRB)

Enable IPX Routing

The first step in enabling IPX routing is to enable it on the router. If you do not specify the node number of the router, the Cisco IOS software uses the hardware Media Access Control (MAC) address currently assigned to it as its node address. This is the MAC address of the first Ethernet, Token Ring, or FDDI interface card.

To enable IPX routing, perform the following global configuration task:

Task	Command
Enable IPX routing.	ipx routing <i>[node]</i>

For an example of how to enable IPX routing, see the “IPX Routing Example” section at the end of this chapter.



Caution If you plan to use DECnet and IPX routing concurrently on the same interface, you should enable DECnet routing first, then enable IPX routing without specifying the optional MAC node number. If you enable IPX before enabling DECnet routing, routing for IPX will be disrupted because DECnet forces a change in the MAC-level node number.

Enable Concurrent Routing and Bridging IRB

You can route IPX on some interfaces and transparently bridge it on other interfaces simultaneously. To do this, you must enable concurrent routing and bridging. To enable concurrent routing and bridging, perform the following task in global configuration mode:

Task	Command
Enable concurrent routing and bridging.	bridge crb ¹

1. This command is documented in the “Transparent Bridging Commands” chapter of the *Bridging and IBM Networking Command Reference*.

Configure Integrated Routing and Bridging (IRB)

IRB enables a user to route IPX traffic between routed interfaces and bridge groups, or route IPX traffic between bridge groups. Specifically, local or unroutable traffic is bridged among the bridged interfaces in the same bridge group. Routable traffic is routed to other routed interfaces or bridge groups. Using IRB, you can do the following:

- Switch packets from a bridged interface to a routed interface
- Switch packets from a routed interface to a bridged interface
- Switch packets within the same bridge group

For more information about configuring integrated routing and bridging, refer to the “Configuring Transparent Bridging” chapter in the *Bridging and IBM Networking Configuration Guide*.

Assign Network Numbers to Individual Interfaces

After you have enabled IPX routing, you assign network numbers to individual interfaces. This enables IPX routing on those interfaces. When you enable IPX routing on an interface, you can also specify an encapsulation (frame type) to use for packets being transmitted on that network.

A single interface can support a single network or multiple logical networks. For a single network, you can configure any encapsulation type. Of course, it should match the encapsulation type of the servers and clients using that network number.

When assigning network numbers to an interface that supports multiple networks, you must specify a different encapsulation type for each network. Because multiple networks share the physical medium, this allows the Cisco IOS software to identify the packets that belong to each network. For example, you can configure up to four IPX networks on a single Ethernet cable, because four encapsulation types are supported for Ethernet. Again, the encapsulation type should match the servers and clients using the same network number.

The following sections describe how to enable IPX routing on interfaces that support a single network and those that support multiple networks.

Assign Network Numbers to Interfaces That Support a Single Network

To assign a network number to an interface that supports a single network, perform the following interface configuration task:

Task	Command
Enable IPX routing on an interface.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>]

If you specify an encapsulation type, be sure to choose the one that matches the one used by the servers and clients on that network.

For an example of how to enable IPX routing, see the “IPX Routing Example” section at the end of this chapter.

Assign Network Numbers to Interfaces That Support Multiple Networks

To assign network numbers to interfaces that support multiple networks, you normally use subinterfaces. A *subinterface* is a mechanism that allows a single physical interface to support multiple logical interfaces or networks. That is, several logical interfaces or networks can be associated with a single hardware interface. Each subinterface must use a distinct encapsulation, and

the encapsulation must match that of the clients and servers using the same network number. To run the NetWare Link Services Protocol (NLSP) on multiple networks on the same physical LAN interface, you must configure subinterfaces.

Any interface configuration parameters that you specify on an individual subinterface are applied to that subinterface only.

To configure multiple IPX networks on a physical interface using subinterfaces, perform the following tasks starting in global configuration mode:

Task	Command
Step 1 Specify a subinterface.	interface <i>type number.subinterface-number</i> ¹
Step 2 Enable IPX routing, specifying the first encapsulation type.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>]

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*.

To configure more than one subinterface, repeat these two steps.

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

For examples of configuring multiple IPX networks on an interface, see the “IPX Routing on Multiple Networks Example” section at the end of this chapter.

Table 5 lists the encapsulation types you can use on IEEE interfaces and shows the correspondence between the encapsulation type and the IPX frame type.

Table 5 Novell IPX Encapsulation Types on IEEE Interfaces

Interface Type	Encapsulation Type	IPX Frame Type
Ethernet	novell-ether (default)	Ethernet_802.3
	arpa	Ethernet_II
	sap	Ethernet_802.2
	snap	Ethernet_Snap
Token Ring	sap (default)	Token-Ring
	snap	Token-Ring_Snap
FDDI	snap (default)	Fddi_Snap
	sap	Fddi_802.2
	novell-fddi	Fddi_Raw

When assigning network numbers to interfaces that support multiple networks, you can also configure primary and secondary networks.

Note In future Cisco IOS software releases, primary and secondary networks will not be supported.

The first logical network you configure on an interface is considered the *primary network*. Any additional networks are considered *secondary networks*. Again, each network on an interface must use a distinct encapsulation and it should match that of the clients and servers using the same network number.

Any interface configuration parameters that you specify on this interface are applied to all the logical networks. For example, if you set the routing update timer to 120 seconds, this value is used on all four networks.

To use primary and secondary networks to configure multiple IPX networks on an interface, perform the following tasks in interface configuration mode:

Task	Command
Step 1 Enable IPX routing on the primary network.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>]
Step 2 Enable IPX routing on a secondary network.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>] [secondary]

To configure more than one secondary network, repeat Step 2 as appropriate.

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

Configure NLSP

The NetWare Link Services Protocol (NLSP) is a link-state routing protocol based on the Open System Interconnection (OSI) Intermediate System to Intermediate System (IS-IS) protocol.

NLSP is designed to be used in a hierarchical routing environment, in which networked systems are grouped into routing areas. Routing areas can then be grouped into routing domains, and domains can be grouped into an internetwork.

Level 1 routers connect networked systems within a given routing area. Areas are connected to each other by Level 2 routers, and domains are connected by Level 3 routers. A Level 2 router also acts as a Level 1 router within its own area; likewise, a Level 3 router also acts as a Level 2 router within its own domain.

The router at each level of the topology stores complete information for its level. For instance, Level 1 routers store complete link-state information about their entire area. This information includes a record of all the routers in the area, the links connecting them, the operational status of the devices and their links, and other related parameters. For each point-to-point link, the database records the end-point devices and the state of the link. For each LAN, the database records which routers are connected to the LAN. Similarly, Level 2 routers would store information about all the areas in the routing domain, and Level 3 routers would store information about all the domains in the internetwork.

Although NLSP is designed for hierarchical routing environments containing Level 1, 2, and 3 routers, only Level 1 routing has been defined in a specification.

Cisco's implementation of NLSP supports the Novell NLSP specification, version 1.1. Our implementation of NLSP also includes read-only NLSP MIB variables.

NLSP is a link-state protocol. This means that every router in a routing area maintains an identical copy of the link-state database, which contains all information about the topology of the area. All routers synchronize their views of the databases among themselves to keep their copies of the link-state databases consistent. NLSP has the following three major databases:

- **Adjacency**—Keeps track of the router's immediate neighbors and the operational status of the directly attached links by exchanging hello packets. Adjacencies are created upon receipt of periodic hello packets. If a link or router goes down, adjacencies time out and are deleted from the database.
- **Link state**—Tracks the connectivity of an entire routing area by aggregating the immediate neighborhood information from all routers into link-state packets (LSPs). LSPs contain lists of adjacencies. They are flooded to all other devices via a reliable flooding algorithm every time a link state changes. LSPs are refreshed every two hours. To keep the size of the link-state database reasonable, NLSP uses fictitious pseudonodes, which represent the LAN as a whole, and designated routers, which originate LSPs on behalf of the pseudonode.
- **Forwarding**—Calculated from the adjacency and link state databases using Dijkstra's shortest path first (SPF) algorithm.

To configure NLSP, you must have configured IPX routing on your router, as described previously in this chapter. Then, you must perform the tasks described in the following sections:

- Define an Internal Network
- Enable NLSP Routing
- Configure NLSP on an Interface

You can optionally perform the tasks described in the following sections:

- Redistribute Routing Information
- Configure RIP and SAP Compatibility
- Configure Maximum Hop Count
- Configure the Link Delay and Throughput
- Configure the Metric Value
- Configure the Priority of the System for Designated Router Election
- Configure Default Routes
- Configure Transmission and Retransmission Intervals
- Log Adjacency State Changes
- Modify Link-State Packet (LSP) Parameters
- Configure Route Aggregation

For an example of enabling NLSP, see the "IPX Routing Protocols Examples" section at the end of this chapter.

Define an Internal Network

An internal network number is an IPX network number assigned to the router. For NLSP to operate, you must configure an internal network number for each device.

To enable IPX routing and to define an internal network number, perform the following tasks in global configuration mode:

Task	Command
Enable IPX routing.	ipx routing
Define an internal network number.	ipx internal-network <i>network-number</i>

Enable NLSP Routing

To enable NLSP, perform the following tasks starting in global configuration mode:

Task	Command
Step 1 Enable NLSP.	ipx router nlspl [<i>tag</i>]
Step 2 Define a set of network numbers to be part of the current NLSP area.	area-address <i>address mask</i>

Configure NLSP on an Interface

You configure NLSP differently on LAN and WAN interfaces, as described in the following sections.

Configure NLSP on a LAN Interface

To configure NLSP on a LAN interface, perform the following tasks in interface configuration mode:

Task	Command
Step 1 Enable IPX routing on an interface.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>]
Step 2 Enable NLSP on the interface.	ipx nlspl [<i>tag</i>] enable

To configure multiple encapsulations on the same physical LAN interfaces, you must configure subinterfaces. Each subinterface must have a different encapsulation type. To do this, perform the following tasks starting in global configuration mode:

Task	Command
Step 1 Specify a subinterface.	interface <i>type number.subinterface-number</i> ¹
Step 2 Enable IPX routing, specifying the first encapsulation type.	ipx network <i>network</i> [encapsulation <i>encapsulation-type</i>]
Step 3 Enable NLSP on the subinterface.	ipx nlspl [<i>tag</i>] enable

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*.

Repeat these three steps for each subinterface.

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

Configure NLSP on a WAN Interface

To configure NLSP on a WAN interface, perform the following tasks starting in global configuration mode:

Task	Command
Step 1 Specify a serial interface.	interface serial <i>number</i> ¹
Step 2 Enable IPXWAN.	ipx ipxwan [<i>local-node unnumbered local-server-name</i> <i>retry-interval retry-limit</i>]
Step 3 Enable NLSP on the interface.	ipx nlspl [<i>tag</i>] enable

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*.

Redistribute Routing Information

Automatic redistribution of one routing protocol into another provides a simple and effective means for building IPX networks in a heterogeneous routing protocol environment. Redistribution is usually effective as soon as you enable an IPX routing protocol. One exception is NLSP and Enhanced IGRP. You must configure the redistribution of Enhanced IGRP into NLSP, and vice versa.

Once you enable Enhanced IGRP and NLSP redistribution, the router makes path decisions based on a predefined, nonconfigurable administrative distance, and prevents redistribution feedback loops without filtering via a stored, external hop count.

To enable redistribution of Enhanced IGRP into NLSP, and vice versa, perform the following tasks, beginning in global configuration mode:

Task	Command
Step 1 Enable NLSP.	ipx router nlspl [<i>tag</i>]
Step 2 From IPX-router configuration mode, enable redistribution of Enhanced IGRP into NLSP.	redistribute eigrpl <i>autonomous-system-number</i>

Task	Command
Step 3 From global configuration mode, enable Enhanced IGRP.	ipx router eigrp <i>autonomous-system-number</i>
Step 4 From IPX-router configuration mode, enable redistribution of NLSP into Enhanced IGRP.	redistribute nlsp [<i>tag</i>]

For an example of how to enable redistribution of Enhanced IGRP and NLSP, see the “Enhanced IGRP and NLSP Route Redistribution Example” section at the end of this chapter.

Configure RIP and SAP Compatibility

Routing Information Protocol (RIP) and SAP are enabled by default on all interfaces configured for IPX, and these interfaces always respond to RIP and SAP requests. When you also enable NLSP on an interface, the interface, by default, generates and sends RIP and SAP periodic traffic only if another RIP router or SAP service is sending RIP or SAP traffic.

To modify the generation of periodic RIP updates on a network enabled for NLSP, perform one of the following tasks in interface configuration mode:

Task	Command
Never generate RIP periodic traffic.	ipx nlsp [<i>tag</i>] rip off
Always generate RIP periodic traffic.	ipx nlsp [<i>tag</i>] rip on
Send RIP periodic traffic only if another RIP router is sending periodic RIP traffic. (This is the default on interfaces configured for NLSP.)	ipx nlsp [<i>tag</i>] rip auto

To modify the generation of periodic SAP updates on a network enabled for NLSP, perform one of the following tasks in interface configuration mode:

Task	Command
Never generate SAP periodic traffic.	ipx nlsp [<i>tag</i>] sap off
Always generate SAP periodic traffic.	ipx nlsp [<i>tag</i>] sap on
Send SAP periodic traffic only if another SAP service is sending periodic SAP traffic. (This is the default on interfaces configured for NLSP.)	ipx nlsp [<i>tag</i>] sap auto

Configure Maximum Hop Count

By default, IPX packets whose hop count exceeds 15 are discarded. In larger internetworks, this may be insufficient. You can increase the hop count to a maximum of 254 hops for Enhanced IGRP and 127 hops for NLSP. To modify the maximum hop count, perform the following task in global configuration mode:

Task	Command
Set the maximum hop count accepted from RIP update packets.	ipx maximum-hops <i>hop</i>

Configure the Link Delay and Throughput

The delay and throughput of each link are used by NLSP as part of its route calculations. By default, these parameters are set to appropriate values or, in the case of IPXWAN, are dynamically measured.

The link delay and throughput you specify replaces the default value or overrides the value measured by IPXWAN when it starts. The value is also supplied to NLSP for use in metric calculations.

To change the link delay, perform the following task in interface configuration mode:

Task	Command
Specify the link delay.	ipx link-delay <i>microseconds</i>

To change the throughput, perform the following task in interface configuration mode:

Task	Command
Specify the throughput.	ipx throughput <i>bits-per-second</i>

Configure the Metric Value

NLSP assigns a default link cost (metric) based on the link throughput. If desired, you can set the link cost manually. To set the NLSP link cost for an interface, perform the following task in interface configuration mode:

Task	Command
Set the metric value for an interface.	ipx nlspl [<i>tag</i>] metric <i>metric-number</i>

Configure the Priority of the System for Designated Router Election

Note In the context of this discussion, the term *designated router* can refer to an access server or a router.

NLSP elects a designated router on each LAN interface. The designated router represents all routers that are connected to the same LAN segment. It creates a virtual router called a *pseudonode*, which generates routing information on behalf of the LAN and transmits it to the remainder of the routing area. The routing information generated includes adjacencies and RIP routes. The use of a designated router significantly reduces the number of entries in the LSP database.

By default, electing a designated router is done automatically. However, you can manually affect the identity of the designated router by changing the priority of the system; the system with the highest priority is elected to be the designated router.

By default, the priority of the system is 44. To change it, perform the following task in interface configuration mode:

Task	Command
Configure the designated router election priority.	ipx nlspl [<i>tag</i>] priority <i>priority-number</i>

Configure Default Routes

The default route is used when a route to any destination network is unknown. By default, IPX treats network number -2 (0xFFFFFFFF) as the default route. To disable the use of this default route, perform the following task in global configuration mode:

Task	Command
Disable default route handling.	no ipx default-route

Unless configured otherwise, all known RIP routes are advertised out each interface. However, you can choose to advertise only the default RIP route if it is known. This greatly reduces the CPU overhead when routing tables are large.

To advertise only the default route via an interface, perform the following task in interface configuration mode:

Task	Command
Advertise only the default route.	ipx advertise-default-route-only <i>network</i>

Configure Transmission and Retransmission Intervals

You can configure the hello transmission interval and holding time multiplier, the complete sequence number PDU (CSNP) transmission interval, the LSP transmission interval, and the LSP retransmission interval.

The hello transmission interval and holding time multiplier used together determine how long a neighboring system should wait after a link or system failure (the “holding time”) before declaring this system to be unreachable. The holding time is equal to the hello transmission interval multiplied by the holding time multiplier.

To configure the hello transmission interval on an interface, perform the following task in interface configuration mode:

Task	Command
Configure the hello transmission interval.	ipx nlsip [<i>tag</i>] hello-interval <i>seconds</i>

To specify the holding time multiplier used on an interface, perform the following task in interface configuration mode:

Task	Command
Configure the hello multiplier.	ipx nlsip [<i>tag</i>] hello-multiplier <i>multiplier</i>

Although not typically necessary, you can configure the CSNP transmission interval. To do so, perform the following task in interface configuration mode:

Task	Command
Configure the CSNP transmission interval.	ipx nlsip [<i>tag</i>] csnp-interval <i>seconds</i>

You can specify how fast LSPs can be flooded out an interface by configuring the LSP transmission interval. To configure the LSP transmission interval, perform the following task in interface configuration mode:

Task	Command
Configure the LSP transmission interval.	ipx nlsp lsp-interval <i>interval</i>

You can set the maximum amount of time that can pass before an LSP will be retransmitted on a WAN link when no acknowledgement is received. To configure this LSP retransmission interval, perform the following task in interface configuration mode:

Task	Command
Configure the LSP retransmission interval.	ipx nlsp [<i>tag</i>] retransmit-interval <i>seconds</i>

Log Adjacency State Changes

You can allow NLSP to generate a log message when an NLSP adjacency changes state (up or down). This may be very useful when monitoring large networks. Messages are logged using the system error message facility. Messages are of the form:

```
%CLNS-5-ADJCHANGE: NLSP: Adjacency to 0000.0000.0034 (Serial0) Up, new adjacency
%CLNS-5-ADJCHANGE: NLSP: Adjacency to 0000.0000.0034 (Serial0) Down, hold time expired
```

To generate log messages when an NLSP adjacency changes state, perform the following task in router configuration mode:

Task	Command
Log NLSP adjacency state changes.	log-adjacency-changes

Modify Link-State Packet (LSP) Parameters

To modify LSP parameters, perform one or more of the following tasks in router configuration mode:

Task	Command
Set the minimum LSP generation interval.	lsp-gen-interval <i>seconds</i>
Set the maximum time the LSP persists.	max-lsp-lifetime <i>seconds</i>
Set the LSP refresh time.	lsp-refresh-interval <i>seconds</i>
Set the maximum size of a link-state packet.	lsp-mtu <i>bytes</i>
Set the minimum time between SPF calculations.	spf-interval <i>seconds</i>

Limit Partial Route Calculations (PRC)

You can control how often the Cisco IOS software performs a partial route (PRC) calculation. Because the PRC calculation is processor-intensive, it may be useful to limit how often this is done, especially on slower router models. Increasing the PRC interval reduces the processor load of the router, but potentially slows down the rate of convergence.

To modify the PRC calculation, perform the following task in router configuration mode:

Task	Command
Set the holddown period between partial route calculations	<code>prc-interval seconds</code>

Configure Route Aggregation

Prior to Cisco IOS Release 11.1, you could segregate IPX internetworks into distinct NLSP areas only by interconnecting them with IPX RIP. With Release 11.1, you can easily do the following tasks:

- Divide large IPX internetworks into multiple NLSP areas
- Redistribute route and service information directly from one NLSP area into other areas
- Enable route summarization

In this document, these independent capabilities are known collectively as the *route aggregation* feature. Cisco has designed the route aggregation feature to be compatible with Novell's *NetWare Link Services Protocol (NLSP) Specification, Revision 1.1*.

Note In the sections that follow, "NLSP version 1.1 routers" refers to routers that support the route aggregation feature, while "NLSP version 1.0 routers" refers to routers that do not. Additionally, all NLSP instances configured on a router running Release 11.1 are NLSP 1.1 instances. They are all capable of generating and using aggregated routes. However, in the text and examples that follow, an NLSP 1.0 instance refers to an instance of NLSP that is in an area that includes NLSP version 1.0 routers.

Benefits

NLSP route summarization provides the following benefits to well-designed IPX networks:

- Compact address representation—A single aggregated route efficiently represents many explicit routes.
- Reduced update bandwidth—Most changes in the explicit routes represented by an aggregated route do not need to be propagated to neighboring areas.
- Reduced computational overhead—Since the routers in one area are unaffected by most changes in adjacent areas, the SPF algorithm runs less often.
- Improved information management—Filtering of route and service information may be done at area boundaries.

As a result, you can build larger IPX networks using route aggregation.

Understanding Area Addresses, Route Summaries, and Aggregated Routes

This section discusses area addresses, route summaries, and aggregated routes. It also describes how area addresses relate to route summaries.

Area Addresses

An *area address* uniquely identifies an NLSP area. The area addresses configured on each router determine the areas to which a router belongs.

An area address consists of a pair of 32-bit hexadecimal numbers that include an area number and a corresponding mask. The mask indicates how much of the area number identifies the area, and how much identifies individual networks in the area. For example, the area address pair *12345600 FFFFFFF00* describes an area composed of 256 networks in the range 12345600 to 123456FF.

You can configure up to three area addresses per NLSP process on the router. Adjacencies are formed only between routers that share at least one common area address.

Route Summaries

A *route summary* defines a set of explicit routes that the router uses to generate an aggregated route. A route summary tells the router how to summarize the set of explicit routes into a single summarized route.

A route summary is similar in form to an area address. That is, the route summary described by *12345600 FFFFFFF00* summarizes the 256 networks in the range 12345600 to 123456FF.

Aggregated Routes

An *aggregated route* is the single, compact data structure that describes many IPX network numbers simultaneously. The aggregated route represents all the explicit routes defined by the route summary. In an LSP, the router expresses an aggregated route as a 1-byte number that gives the length, in bits, of the portion of the 32-bit network number common to all summarized addresses. The aggregated route for *12345600 FFFFFFF00* is *18 12345600*.

Relationship Between Area Addresses and Route Summaries

When you enable route summarization in Release 11.1 while running multiple instances of NLSP, the router performs default route summarization based on the area address configured in each NLSP area. That is, explicit routes that match the area address in a given area are not redistributed individually into neighboring NLSP areas. Instead, the router redistributes a single aggregated route that is equivalent to the area address into neighboring areas.

Understanding NLSP Areas

This section describes single versus multiple NLSP areas and discusses the router's behavior when you mix NLSP versions within a single NLSP area.

Single Versus Multiple NLSP Areas

NLSP version 1.0 routers support only a single, Level 1 area. Two routers form an adjacency only if they share at least one configured area address in common. The union of routers with adjacencies in common form an area.

Each router within the NLSP area has its own adjacencies, link-state, and forwarding databases. Further, each router's link-state database is identical. Within the router, these databases operate collectively as a single *process* or *instance* to discover, select, and maintain route information about the area. NLSP version 1.0 routers and NLSP version 1.1 routers that exist within a single area use a single NLSP instance.

With NLSP version 1.1 and Cisco IOS Release 11.1, multiple instances of NLSP may exist on a given router. Each instance discovers, selects, and maintains route information for a separate NLSP area. Each instance has its own copy of the NLSP adjacency and link state database for its area. However, all instances (along with other routing protocols such as RIP and Enhanced IGRP) share a single copy of the forwarding table.

Mixing NLSP Versions in a Single Area

You can have NLSP version 1.1 routers and NLSP version 1.0 routers in the same area. However, NLSP version 1.0 routers do not recognize aggregated routes. For this reason, the default behavior of Cisco IOS Release 11.1 software is to not generate aggregated routes. To prevent routing loops in a mixed environment, packets routed via an aggregated route by an NLSP version 1.1 router are dropped if the next hop is an NLSP version 1.0 router.

Note In general, you should ensure that all routers in an area are running NLSP version 1.1-capable software before you enable route summarization on any of the routers in an area.

Understanding Route Redistribution

Because you can configure multiple NLSP areas, you must understand how the router passes route information from one area to another. Passing route information from one area to another, or from one protocol to another, is known as *route redistribution*. Additionally, you must understand the router's default route redistribution behavior before configuring route summarization.

This section describes the default route redistribution behavior between multiple NLSP areas, between NLSP and Enhanced IGRP, and between NLSP and RIP.

Default Redistribution Between Multiple NLSP Areas

Regardless of the NLSP version, Cisco IOS Release 11.1 redistributes routes between multiple NLSP areas by default. That is, redistribution between multiple NLSP version 1.1 areas, between multiple NLSP version 1.0 areas, and between NLSP version 1.1 and NLSP version 1.0 areas is enabled by default. All routes are redistributed as individual, explicit routes.

Default Redistribution Between NLSP and Enhanced IGRP

Route redistribution between instances of NLSP (version 1.1 or version 1.0) and Enhanced IGRP is disabled by default. You must explicitly configure this type of redistribution. Refer to the “Redistribute Routing Information” section in this chapter for information about configuring redistribution between NLSP and Enhanced IGRP.

Default Redistribution Between NLSP and RIP

Route redistribution between instances of NLSP (version 1.1 or version 1.0) and RIP is enabled by default. All routes are redistributed as individual, explicit routes.

Understanding Route Summarization

Route summarization is disabled by default to avoid the generation of aggregated routes in an area running mixed versions of NLSP. You can explicitly enable route summarization on a router running Cisco IOS Release 11.1. This section describes default route summarization, customized route summarization, and the relationship between filtering and route summarization.

Default Route Summarization

When you explicitly enable route summarization, the default route summarization depends on the the following circumstances:

- All routers use NLSP version 1.1—The area address for each NLSP instance is used as the basis for generating aggregated routes.
- Some routers use NLSP version 1.1 and some use NLSP version 1.0—The area address for each NLSP instance is used as the basis for generating aggregated routes; however, NLSP version 1.0 routers do not recognize aggregated routes. You must not enable route aggregation on the NLSP version 1.0 instance, or you must configure customized route summarization to prevent generation of aggregated routes from the NLSP version 1.0 areas. See the “Customized Route Summarization” section.
- Some routers use Enhanced IGRP and NLSP version 1.1—There is no default route summarization. You must configure customized route summarization to generate aggregated routes from Enhanced IGRP to NLSP version 1.1. See the “Customized Route Summarization” section.
- Some routers use RIP and NLSP version 1.1—There is no default route summarization. You must configure customized route summarization to generate aggregated routes from RIP to NLSP version 1.1. See the “Customized Route Summarization” section.

In the case of the first two circumstances, the area address for each NLSP instance is used as the basis for generating aggregated routes. That is, all explicit routes that match a local area address generate a common aggregated route. The router redistributes only the aggregated route into other NLSP areas; explicit routes (and more specific aggregated routes) represented by a particular aggregated route are filtered.

Note The router continues to redistribute into other areas the explicit routes that do *not* match the area address.

Customized Route Summarization

You can also customize the router's route summarization behavior using the **redistribute** IPX-router subcommand with an access list. The access list specifies in detail which routes to summarize and which routes to redistribute explicitly. In this case, the router ignores area addresses and uses only the access list as a template to control summarization and redistribution.

In addition, you must use customized route summarization in environments that use either of the following combinations:

- Enhanced IGRP and NLSP version 1.1
- RIP and NLSP version 1.1

Route summarization between Enhanced IGRP and NLSP is controlled by the access list. Route summarization is possible only in the Enhanced IGRP-to-NLSP direction. Routes redistributed from NLSP to Enhanced IGRP are always explicit routes.

Route summarization between RIP and NLSP is also controlled by the access list. Route summarization is possible only in the RIP-to-NLSP direction. Routes redistributed from NLSP to RIP are always explicit routes. Use the default route instead to minimize routing update overhead, yet maximize reachability in a RIP-only area.

Note Before introducing the default route into a RIP-only area, be sure that all routers and servers in the area are upgraded to understand and use the default route.

In a well-designed network, within each NLSP area, most external networks are reachable by a few aggregated routes, while all other external networks are reachable either by individual explicit routes or by the default route.

Relationship Between Filtering and Route Summarization

Redistribution of routes and services into and out of an NLSP area may be modified using filters. Filters are available for both input and output directions. Refer to the **distribute-list in**, **distribute-list out**, **distribute-sap-list in**, and **distribute-sap-list out** commands in the *Network Protocols Command Reference, Part 2* for more information on these filters.

Filtering is independent of route summarization, but may affect it indirectly, since filters are always applied before the aggregation algorithm is applied. It is possible to filter all explicit routes that could generate aggregated routes, making the router unable to generate aggregated routes even though route aggregation is turned on.

Understanding Service and Path Selection

The router always accepts service information as long as the service’s network is reachable by an explicit route, an aggregated route, or the default route. When choosing a server for a Get Nearest Server (GNS) response, the tick value of the route to each eligible server is used as the metric. No distinction is made between explicit and summary routes in this determination. If the tick values are equal, then the hop count is used as a tiebreaker. However, because there is no hop value associated with an aggregated route, services reachable via an explicit route are always preferred over those reachable via only an aggregated route.

An NLSP version 1.1 router always uses the most explicit match to route packets. That is, the router always uses an explicit route if possible. If not, then a matching aggregated route is used. If multiple aggregated routes match, then the most explicit (longest match) is used. If no aggregated route is present, then the default route is used as a last resort.

Configure Route Aggregation Task List

To configure the route aggregation feature, perform one or more of the following tasks:

- Configure Route Aggregation for Multiple NLSP Version 1.1 Areas.
- Configure Route Aggregation for NLSP Version 1.1 and NLSP Version 1.0 Areas.
- Configure Route Aggregation for Enhanced IGRP and NLSP Version 1.1 Environments.
- Configure Route Aggregation for RIP and NLSP Version 1.1 Environments.

Configure Route Aggregation for Multiple NLSP Version 1.1 Areas

Redistribution between multiple NLSP 1.1 areas is enabled by default. Because multiple NLSP processes are present on the router, a *tag* or label identifies each. For each instance, configure an appropriate area address and, optionally, enable route summarization. Finally, enable NLSP on appropriate interfaces. Be sure to use the correct tag (process) identifier to associate that interface with the appropriate NLSP area.

Note Note that the tag used to identify an NLSP instance is meaningful only locally within the router. NLSP adjacencies and areas are determined by the area address and interfaces configured for each instance of NLSP running on each router. There is no need (other than administrative convenience) to ensure that individual tags match between routers.

To configure the route aggregation feature with the default route summarization behavior, perform these steps for each NLSP process:

Task	Command
Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nls [<i>tag</i>]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address <i>address mask</i>
Step 3 (Optional) From router configuration mode, enable route summarization.	route-aggregation

Task	Command
Step 4 From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [<i>tag</i>] enable

For an example of how to configure this type of route aggregation, see “NLSP Route Aggregation for NLSP Version 1.1 and Version 1.0 Areas Example” section at the end of this chapter.

To configure the route aggregation feature with customized route summarization behavior, perform these steps for each NLSP process:

Task	Command
Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nlsip [<i>tag</i>]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address <i>address mask</i>
Step 3 Enable route summarization from router configuration mode.	route-aggregation
Step 4 From router configuration mode, use the redistribute command with an access list in the range of 1200 to 1299. In this case, the <i>tag</i> argument identifies a unique NLSP process.	redistribute nlsip [<i>tag</i>] access-list <i>access-list-number</i>
Step 5 From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [<i>tag</i>] enable
Step 6 From global configuration mode, define the access list to redistribute an aggregated route instead of the explicit route. For each address range you want to summarize, use the deny keyword.	access-list <i>access-list-number</i> deny <i>network network-mask</i> [ticks ticks] [area-count area-count]
Step 7 (Optional) Terminate the access list with a “permit all” statement to redistribute all other routes as explicit routes.	access-list <i>access-list-number</i> permit -1

Configure Route Aggregation for NLSP Version 1.1 and NLSP Version 1.0 Areas

By default, redistribution is enabled between multiple instances of NLSP. Route summarization, when enabled, is possible in one direction only—from NLSP version 1.0 to NLSP version 1.1.

To configure the route aggregation feature with default route summarization behavior, perform the following steps for each NLSP process:

Task	Command
Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nlsip [<i>tag</i>]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address <i>address mask</i>

Task	Command
Step 3 For NLSP version 1.1 areas, enable route summarization from router configuration mode. Skip this step for NLSP version 1.0 areas.	route-aggregation
Step 4 From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [tag] enable

To configure the route aggregation feature with customized route summarization behavior, perform the tasks in the following two tables.

For the NLSP version 1.1 process, perform these steps:

Task	Command
Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nlsip [tag]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address address mask
Step 3 For NLSP version 1.1 areas, enable route summarization from router configuration mode.	route-aggregation
Step 4 (Optional) From router configuration mode, redistribute NLSP version 1.0 into the NLSP version 1.1 area. Include an access list number between 1200 and 1299.	redistribute nlsip [tag] access-list access-list-number
Step 5 From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [tag] enable
Step 6 (Optional) From global configuration mode, define the access list to redistribute an aggregated route instead of explicit routes learned from the NLSP version 1.0 area. For each address range you want to summarize, use the deny keyword.	access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]
Step 7 (Optional) Terminate the access list with a “permit all” statement to redistribute all other routes as explicit routes.	access-list access-list-number permit -1

For the NLSP version 1.0 process, perform these steps:

Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nlsip [tag]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address address mask

Step 3	From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [tag] enable
---------------	--	-------------------------------

For an example of how to configure the route aggregation feature with this type of customized route summarization, refer to the “NLSP Route Aggregation for NLSP Version 1.1 and Version 1.0 Areas Example” section at the end of this chapter.

Configure Route Aggregation for Enhanced IGRP and NLSP Version 1.1 Environments

Redistribution is not enabled by default. Additionally, summarization is possible in the Enhanced IGRP to NLSP direction only.

For each NLSP version 1.1 process, perform these steps, beginning in global configuration mode:

Task	Command
Step 1 Enable NLSP routing and identify the process with a unique tag.	ipx router nlsip [tag]
Step 2 From router configuration mode, define up to three area addresses for the process.	area-address address mask
Step 3 (Optional) From router configuration mode, enable route summarization.	route-aggregation
Step 4 (Optional) From router configuration mode, redistribute Enhanced IGRP into the NLSP version 1.1 area. Include an access list number between 1200 and 1299.	redistribute {eigrp autonomous-system-number} [access-list access-list-number]
Step 5 From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.	ipx nlsip [tag] enable
Step 6 (Optional) From global configuration mode, define the access list to redistribute an aggregated route instead of explicit routes learned from Enhanced IGRP. For each address range you want to summarize, use the deny keyword.	access-list access-list-number deny network network-mask [ticks ticks] [area-count area-count]
Step 7 (Optional) Terminate the access list with a “permit all” statement to redistribute all other Enhanced IGRP routes as explicit routes.	access-list access-list-number permit -1

For each Enhanced IGRP autonomous system, perform these steps, beginning in global configuration mode:

Task	Command
Step 1 Enable Enhanced IGRP.	ipx router eigrp autonomous-system-number
Step 2 From router configuration mode, specify the networks to be enabled for Enhanced IGRP.	network {network-number all}

Step 3	From router configuration mode, redistribute NLSP version 1.1 into Enhanced IGRP.	redistribute nlsp <i>[tag]</i>
---------------	---	---------------------------------------

For an example of how to configure this type of route aggregation, refer to the “NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example” section at the end of this chapter.

Configure Route Aggregation for RIP and NLSP Version 1.1 Environments

Because redistribution between RIP and NLSP is enabled by default, you only need to enable the route summarization, if desired, to configure all the capabilities of the route aggregation feature.

For each NLSP version 1.1 process, perform these steps, beginning in global configuration mode:

Task	Command
Step 1	Enable NLSP routing and identify the process with a unique tag.
Step 2	From router configuration mode, define up to three area addresses for the process.
Step 3	(Optional) From router configuration mode, enable route summarization.
Step 4	(Optional) From router configuration mode, redistribute RIP routes into the NLSP version 1.1 area. Include an access list number between 1200 and 1299.
Step 5	From interface configuration mode, enable NLSP on each network in the area described by the <i>tag</i> argument.
Step 6	(Optional) From global configuration mode, define the access list to redistribute an aggregated route instead of explicit RIP routes. For each address range you want to summarize, use the deny keyword.
Step 7	(Optional) Terminate the access list with a “permit all” statement to redistribute all other RIP routes as explicit routes.

For an example of how to configure this type of route aggregation, refer to the “NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example” section at the end of this chapter.

Configure IPX Enhanced IGRP

Enhanced IGRP is an enhanced version of the Interior Gateway Routing Protocol (IGRP) developed by Cisco Systems, Inc. Enhanced IGRP uses the same distance vector algorithm and distance information as IGRP. However, the convergence properties and the operating efficiency of Enhanced IGRP have improved significantly over IGRP.

The convergence technology is based on research conducted at SRI International and employs an algorithm referred to as the Diffusing Update Algorithm (DUAL). This algorithm guarantees loop-free operation at every instant throughout a route computation, and allows all routers involved in a topology change to synchronize at the same time. Routers that are not affected by topology changes are not involved in recomputations. The convergence time with DUAL rivals that of any other existing routing protocol.

Enhanced IGRP offers the following features:

- **Fast convergence**—The DUAL algorithm allows routing information to converge as quickly as any currently available routing protocol.
- **Partial updates**—Enhanced IGRP sends incremental updates when the state of a destination changes, instead of sending the entire contents of the routing table. This feature minimizes the bandwidth required for Enhanced IGRP packets.
- **Less CPU usage than IGRP**—This occurs because full update packets do not have to be processed each time they are received.
- **Neighbor discovery mechanism**—This is a simple hello mechanism used to learn about neighboring routers. It is protocol-independent.
- **Scaling**—Enhanced IGRP scales to large networks.

Enhanced IGRP has the following four basic components:

- Neighbor discovery/recovery
- Reliable transport protocol
- DUAL finite-state machine
- Protocol-dependent modules

Neighbor discovery/recovery is the process that routers use to dynamically learn of other routers on their directly attached networks. Routers must also discover when their neighbors become unreachable or inoperative. Neighbor discovery/recovery is achieved with low overhead by periodically sending small hello packets. As long as hello packets are received, a router can determine that a neighbor is alive and functioning. Once this status is determined, the neighboring devices can exchange routing information.

The reliable transport protocol is responsible for guaranteed, ordered delivery of Enhanced IGRP packets to all neighbors. It supports intermixed transmission of multicast and unicast packets. Some Enhanced IGRP packets must be transmitted reliably, and others need not be. For efficiency, reliability is provided only when necessary. For example, on a multiaccess network that has multicast capabilities (such as Ethernet) it is not necessary to send hellos reliably to all neighbors individually. Therefore, Enhanced IGRP sends a single multicast hello with an indication in the packet informing the receivers that the packet need not be acknowledged. Other types of packets (such as updates) require acknowledgment, and this is indicated in the packet. The reliable transport has a provision to send multicast packets quickly when there are unacknowledged packets pending. Doing so helps ensure that convergence time remains low in the presence of varying speed links.

The DUAL finite-state machine embodies the decision process for all route computations. It tracks all routes advertised by all neighbors. DUAL uses the distance information (known as a metric) to select efficient, loop-free paths. DUAL selects routes to be inserted into a routing table based on

feasible successors. A *successor* is a neighboring router used for packet forwarding that has a least-cost path to a destination that is guaranteed not to be part of a routing loop. When there are no feasible successors but there are neighbors advertising the destination, a recomputation must occur. This is the process whereby a new successor is determined. The amount of time it takes to recompute the route affects the convergence time. Recomputation is processor-intensive. It is advantageous to avoid recomputation if it is not necessary. When a topology change occurs, DUAL will test for feasible successors. If there are feasible successors, it will use any it finds in order to avoid unnecessary recomputation.

The protocol-dependent modules are responsible for network layer protocol-specific tasks. They are also responsible for parsing Enhanced IGRP packets and informing DUAL of the new information received. Enhanced IGRP asks DUAL to make routing decisions, but the results are stored in the IPX routing table. Also, Enhanced IGRP is responsible for redistributing routes learned by other IPX routing protocols.

To enable IPX Enhanced IGRP, complete the tasks in the following sections. Only the first task is required; the remaining task is optional.

- Enable IPX Enhanced IGRP
- Configure Miscellaneous Enhanced IGRP Parameters

Enable IPX Enhanced IGRP

To create an IPX Enhanced IGRP routing process, perform the following tasks:

Task	Command
Step 1 Enable an Enhanced IGRP routing process in global configuration mode.	ipx router eigrp <i>autonomous-system-number</i>
Step 2 Enable Enhanced IGRP on a network in IPX router configuration mode.	network { <i>network-number</i> all }

To associate multiple networks with an Enhanced IGRP routing process, you can repeat Step 2.

For an example of how to enable Enhanced IGRP, see the “IPX Enhanced IGRP Example” section at the end of this chapter.

Configure Miscellaneous Enhanced IGRP Parameters

To configure the following miscellaneous Enhanced IGRP parameters, perform one or more of the tasks described in the following sections:

- Redistribute Routing Information
- Adjust the Interval between Hello Packets and the Hold Time
- Disable Split Horizon
- Control SAP Updates
- Control the Advertising of Routes in Routing Updates
- Control the Processing of Routing Updates
- Control the Advertising of Services in SAP Updates
- Control the Processing of SAP Updates
- Query the Backup Server

- Log Enhanced IGRP Neighbor Adjacency Changes
- Configure the Percentage of Link Bandwidth Used by Enhanced IGRP

Redistribute Routing Information

By default, the Cisco IOS software redistributes IPX RIP routes into Enhanced IGRP, and vice versa.

To disable route redistribution, perform the following task in IPX router configuration mode:

Task	Command
Disable redistribution of RIP routes into Enhanced IGRP and Enhanced IGRP routes into RIP.	no redistribute {rip eigrp <i>autonomous-system-number</i> connected static}

The Cisco IOS software does not automatically redistribute NLSP routes into Enhanced IGRP routes and vice versa. You must configure this type of redistribution. To do so, perform the following tasks, beginning in global configuration mode:

Task	Command
Step 1 From global configuration mode, enable Enhanced IGRP.	ipx router eigrp <i>autonomous-system-number</i>
Step 2 From IPX-router configuration mode, enable redistribution of NLSP into Enhanced IGRP.	redistribute nlsip [<i>tag</i>]
Step 3 Enable NLSP.	ipx router nlsip [<i>tag</i>]
Step 4 From IPX-router configuration mode, enable redistribution of Enhanced IGRP into NLSP.	redistribute eigrp <i>autonomous-system-number</i>

For an example of how to enable redistribution of Enhanced IGRP and NLSP, see the “Enhanced IGRP and NLSP Route Redistribution Example” section at the end of this chapter.

Adjust the Interval between Hello Packets and the Hold Time

You can adjust the interval between hello packets and the hold time.

Routers periodically send hello packets to each other to dynamically learn of other devices on their directly attached networks. Routers use this information to discover who their neighbors are and to discover when their neighbors become unreachable or inoperative.

By default, hello packets are sent every 5 seconds. The exception is on low-speed, nonbroadcast, multiaccess (NBMA) media, where the default hello interval is 60 seconds. Low speed is considered to be a rate of T1 or slower, as specified with the bandwidth interface configuration command. The default hello interval remains 5 seconds for high-speed NBMA networks. Note that for the purposes of Enhanced IGRP, Frame-relay and SMDS networks may or may not be considered to be NBMA. These networks are considered NBMA if the interface has not been configured to use physical multicasting; otherwise they are considered not to be NBMA.

You can configure the hold time on a specified interface for a particular Enhanced IGRP routing process designated by the autonomous system number. The hold time is advertised in hello packets and indicates to neighbors the length of time they should consider the sender valid. The default hold time is 3 times the hello interval, or 15 seconds.

To change the interval between hello packets, perform the following task in interface configuration mode:

Task	Command
Set the interval between hello packets.	ipx hello-interval eigrp <i>autonomous-system-number seconds</i>

On very congested and large networks, 15 seconds may not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you may want to increase the hold time. To do this, perform the following task in interface configuration mode:

Task	Command
Set the hold time.	ipx hold-time eigrp <i>autonomous-system-number seconds</i>

Note Do not adjust the hold time without consulting with Cisco technical support.

Disable Split Horizon

Split horizon controls the sending of Enhanced IGRP update and query packets. If split horizon is enabled on an interface, these packets are not sent for destinations if this interface is the next hop to that destination.

By default, split horizon is enabled on all interfaces.

Split horizon blocks information about routes from being advertised by the Cisco IOS software out any interface from which that information originated. This behavior usually optimizes communication among multiple routers, particularly when links are broken. However, with nonbroadcast networks (such as Frame Relay and SMDs), situations can arise for which this behavior is less than ideal. For these situations, you can disable split horizon.

To disable split horizon, perform the following task in interface configuration mode:

Task	Command
Disable split horizon.	no ipx split-horizon eigrp <i>autonomous-system-number</i>

Control SAP Updates

If IPX Enhanced IGRP peers are found on an interface, you can configure the Cisco IOS software to send SAP updates either periodically or when a change occurs in the SAP table. When no IPX Enhanced IGRP peer is present on the interface, periodic SAPs are always sent.

On serial lines, by default, if an Enhanced IGRP neighbor is present, the Cisco IOS software sends SAP updates only when the SAP table changes. On Ethernet, Token Ring, and FDDI interfaces, by default, the software sends SAP updates periodically. To reduce the amount of bandwidth required to send SAP updates, you might want to disable the periodic sending of SAP updates on LAN interfaces. Do this only when all nodes out this interface are Enhanced IGRP peers; otherwise, loss of SAP information on the other nodes will result.

To send SAP updates only when a change occurs in the SAP table, perform the following task in interface configuration mode:

Task	Command
Send SAP updates only when a change in the SAP table occurs, and send SAP changes only.	ipx sap-incremental eigrp <i>autonomous-system-number</i> rsup-only

To send periodic SAP updates, perform the following task in interface configuration mode:

Task	Command
Send SAP updates periodically.	no ipx sap-incremental eigrp <i>autonomous-system-number</i>

For an example of how to configure SAP updates, see the “Enhanced IGRP SAP Update Examples” section at the end of this chapter.

Control the Advertising of Routes in Routing Updates

To control which devices learn about routes, you can control the advertising of routes in routing updates. To do this, perform the following task in router configuration mode:

Task	Command
Control the advertising of routes in routing updates.	distribute-list <i>access-list-number</i> out [<i>interface-name</i> <i>routing-process</i>]

Control the Processing of Routing Updates

To control the processing of routes listed in incoming updates, perform the following task in router configuration mode:

Task	Command
Control which incoming route updates are processed.	distribute-list <i>access-list-number</i> in [<i>interface-name</i>]

Control the Advertising of Services in SAP Updates

To control which devices learn about services, you can control the advertising of these services in SAP updates. To do this, perform the following task in router configuration mode:

Task	Command
Control the advertising of services in SAP updates.	distribute-sap-list <i>access-list-number</i> out [<i>interface-name</i> <i>routing-process</i>]

For a configuration example of controlling the advertisement of SAP updates, see the “Advertisement and Processing of SAP Update Examples” section at the end of this chapter.

Control the Processing of SAP Updates

To control the processing of routes listed in incoming updates, perform the following task in router configuration mode:

Task	Command
Control which incoming SAP updates are processed.	distribute-sap-list <i>access-list-number</i> in <i>[interface-name]</i>

For a configuration example of controlling the processing of SAP updates, see the “Advertisement and Processing of SAP Update Examples” section at the end of this chapter.

Query the Backup Server

The backup server table is a table kept for each Enhanced IGRP peer. It lists the IPX servers that have been advertised by that peer. If a server is removed from the main server table at any time and for any reason, the Cisco IOS software examines the backup server table to see if this just-removed server is known by any of the Enhanced IGRP peers. If it is, the information from that peer is advertised back into the main server table just as if that peer had readvertised the server information to this router. Using this method to allow the router to keep the backup server table consistent with what is advertised by each peer means that only changes to the table must be advertised between Enhanced IGRP routers; full periodic updates do not need to be sent.

By default, the Cisco IOS software queries its own copy of each Enhanced IGRP neighbor’s backup server table every 15 seconds. To change this interval, perform the following task in global configuration mode:

Task	Command
Specify the minimum period of time between successive queries of a neighbor’s backup server table.	ipx backup-server-query-interval <i>interval</i>

Log Enhanced IGRP Neighbor Adjacency Changes

You can enable the logging of neighbor adjacency changes to monitor the stability of the routing system and to help you detect problems. By default, adjacency changes are not logged.

To enable logging of Enhanced IGRP neighbor adjacency changes, perform the following task in global configuration mode:

Task	Command
Enable logging of Enhanced IGRP neighbor adjacency changes.	log-neighbor-changes

Configure the Percentage of Link Bandwidth Used by Enhanced IGRP

By default, Enhanced IGRP packets consume a maximum of 50 percent of the link bandwidth, as configured with the **bandwidth** interface subcommand. If a different value is desired, use the **ipx eigrp-bandwidth-percent** command. This command may be useful if a different level of link utilization is required, or if the configured bandwidth does not match the actual link bandwidth (it may have been configured to influence route metric calculations).

To configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface, perform the following task in interface configuration mode:

Task	Command
Configure the percentage of bandwidth that may be used by Enhanced IGRP on an interface. <code>ipx enhanced igrp.</code>	<code>ipx eigrp-bandwidth-percent percent</code>

For an example of how to configure the percentage of Enhanced IGRP bandwidth, see the “IPX Enhanced IGRP Bandwidth Configuration Example” section at the end of this chapter.

Control Access to IPX Networks

To control access to IPX networks, you create access lists and then apply them with filters to individual interfaces.

The following are the four types of IPX access lists that you can use to filter various kinds of traffic:

- Standard access list—Restricts traffic based on the source network number. You can further restrict traffic by specifying a destination address and a source and destination address mask. Standard IPX access lists have numbers from 800 to 899.
- Extended access list—Restricts traffic based on the IPX protocol type. You can further restrict traffic by specifying source and destination addresses and address masks, and source and destination sockets. Extended IPX access lists have numbers from 900 to 999.
- SAP access list—Restricts traffic based on the IPX Service Advertising Protocol (SAP) type. These lists are used for SAP filters and Get Nearest Server (GNS) response filters. Novell SAP access lists have numbers from 1,000 to 1,099.
- IPX NetBIOS access list—Restricts IPX NetBIOS traffic based on NetBIOS names, not numbers.

There are 13 different IPX filters that you can define for IPX interfaces. They fall into the following five groups:

- Generic filters—Control which data packets are routed in or out of an interface based on the packet’s source and destination addresses and IPX protocol type.
- Routing table filters—Control which Routing Information Protocol (RIP) updates are accepted and advertised by the Cisco IOS software, and from which devices the local router accepts RIP updates.
- SAP filters—Control which SAP services the Cisco IOS software accepts and advertises and which Get Nearest Server (GNS) response messages it sends out.
- IPX NetBIOS filters—Control incoming and outgoing IPX NetBIOS packets.
- Broadcast filters—Control which broadcast packets are forwarded.

Table 6 summarizes the filters and the commands you use to define them. Use the **show ipx interfaces** command to display the filters defined on an interface.

Table 6 IPX Filters

Filter Type	Command Used to Define Filter
Generic filters	
Filter inbound or outbound packets based on the contents of the IPX network header.	ipx access-group <i>access-list-number</i> [in out]
Routing table filters	
Control which networks are added to the routing table.	ipx input-network-filter <i>access-list-number</i>
Control which networks are advertised in routing updates.	ipx output-network-filter <i>access-list-number</i>
Control the routers from which updates are accepted.	ipx router-filter <i>access-list-number</i>
SAP filters	
Filter incoming service advertisements.	ipx input-sap-filter <i>access-list-number</i>
Filter outgoing service advertisements.	ipx output-sap-filter <i>access-list-number</i>
Control the routers from which SAP updates are accepted.	ipx router-sap-filter <i>access-list-number</i>
Filter list of servers in GNS response messages.	ipx output-gns-filter <i>access-list-number</i>
IPX NetBIOS filters	
Filter incoming packets by node name.	ipx netbios input-access-filter host <i>name</i>
Filter incoming packets by byte pattern.	ipx netbios input-access-filter bytes <i>name</i>
Filter outgoing packets by node name.	ipx netbios output-access-filter host <i>name</i>
Filter outgoing packets by byte pattern.	ipx netbios output-access-filter bytes <i>name</i>
Broadcast filters	
Control which broadcast packets are forwarded.	ipx helper-list <i>access-list-number</i>

Keep the following in mind when configuring IPX network access control:

- Access lists entries are scanned in the order you enter them. The first matching entry is used. To improve performance, it is recommended that you place the most commonly used entries near the beginning of the access list.
- An implicit *deny everything* entry is defined at the end of an access list unless you include an explicit *permit everything* entry at the end of the list.
- All new entries to an existing list are placed at the end of the list. You cannot add an entry to the middle of a list. This means that if you have previously included an explicit *permit everything* entry, new entries will never be scanned. The solution is to delete the access list and re-enter it with the new entries.
- Do not to set up conditions that result in packets getting lost. One way this can happen is when a device or interface is configured to advertise services on a network that has access lists that deny these packets.
- You cannot filter within an NLSP area. You can filter at the boundary of NLSP and RIP or SAP, though restrictions do apply. For more information about filtering, refer to the *Novell NetWare Link Services Protocol (NLSP) Specification*.

You perform the tasks in one or more of the following sections to control access to IPX networks:

- Create Access Lists
- Create Generic Filters
- Create Filters for Updating the Routing Table
- Create SAP Filters
- Create GNS Response Filters
- Create IPX NetBIOS Filters
- Create Broadcast Message Filters

Create Access Lists

To create access lists, you can perform one or more of the following tasks in global configuration mode:

Task	Command
Create a standard IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>source-network</i> [. <i>source-node</i> [<i>source-node-mask</i>]] [<i>destination-network</i> [. <i>destination-node</i> [<i>destination-node-mask</i>]]]
Create an extended IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> [<i>source-network</i> [. <i>source-node</i> [<i>source-network-mask</i> , <i>source-node-mask</i>]]] <i>source-socket</i> [<i>destination-network</i> [<i>destination-node</i> [<i>destination-network-mask</i> , <i>destination-node-mask</i>] <i>destination-socket</i>] [log]
Create an IPX access list for SAP filters.	access-list <i>access-list-number</i> { deny permit } <i>network</i> [. <i>node</i>] [<i>network-mask</i> <i>node-mask</i>] [<i>service-type</i> [<i>server-name</i>]]

Task	Command
Create an access list for filtering IPX NetBIOS packets by node name.	netbios access-list host <i>name</i> {deny permit} <i>string</i>
Create an access list for filtering IPX NetBIOS packets by arbitrary byte pattern.	netbios access-list bytes <i>name</i> {deny permit} <i>offset</i> <i>byte-pattern</i>

Once you have created an access list, apply it to a filter on the appropriate interfaces as described in the sections that follow. This activates the access list.

Create Generic Filters

Generic filters determine which data packets to receive from or send to an interface, based on the packet's source and destination addresses, IPX protocol type, and source and destination socket numbers.

To create generic filters, perform the following tasks:

Step 1 Create a standard or an extended access list.

Step 2 Apply a filter to an interface.

To create an access list, perform one of the following tasks in global configuration mode:

Task	Command
Create a standard IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> {deny permit} <i>source-network</i> [. <i>source-node</i>] <i>source-node-mask</i>] [<i>destination-network</i> [. <i>destination-node</i> <i>destination-node-mask</i>]]
Create an extended IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> {deny permit} <i>protocol</i> [<i>source-network</i> [. <i>source-node</i> <i>source-network-mask</i> , <i>source-node-mask</i>]] <i>source-socket</i> [<i>destination-network</i> [<i>destination-node</i> <i>destination-network-mask</i> , <i>destination-node-mask</i>]] <i>destination-socket</i>][log]

To apply a generic filter to an interface, perform the following task in interface configuration mode:

Task	Command
Apply a generic filter to an interface.	ipx access-group <i>access-list-number</i> [in out]

You can apply only one input filter and one output filter per interface or subinterface. You cannot configure an output filter on an interface where autonomous switching is already configured. Similarly, you cannot configure autonomous switching on an interface where an output filter is already present. You cannot configure an input filter on an interface if autonomous switching is already configured on *any* interface. Likewise, you cannot configure input filters if autonomous switching is already enabled on *any* interface.

For an example of creating a generic filter, see the “IPX Network Access Example” section at the end of this chapter.

Create Filters for Updating the Routing Table

Routing table update filters control the entries that the Cisco IOS software accepts for its routing table, and the networks that it advertises in its routing updates.

To create filters to control updating of the routing table, perform the following tasks:

Step 1 Create a standard or an extended access list.

Step 2 Apply one or more routing filters to an interface.

To create an access list, perform one of the following tasks in global configuration mode:

Task	Command
Create a standard IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>source-network</i> [<i>.source-node</i> [<i>source-node-mask</i>]] [<i>destination-network</i> [<i>.destination-node</i> [<i>destination-node-mask</i>]]]
Create an extended IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> [<i>source-network</i> [<i>.source-node</i> [<i>source-network-mask</i> , <i>source-node-mask</i>]]] <i>source-socket</i> [<i>destination-network</i> [<i>.destination-node</i> [<i>destination-network-mask</i> , <i>destination-node-mask</i>]]] [<i>destination-socket</i>]

To apply routing table update filters to an interface, perform one or more of the following tasks in interface configuration mode:

Task	Command
Control which networks are added to the routing table when IPX routing updates are received.	ipx input-network-filter <i>access-list-number</i>
Control which networks are advertised in RIP routing updates sent out by the Cisco IOS software.	ipx output-network-filter <i>access-list-number</i>
Control which networks are advertised in the Enhanced IGRP routing updates sent out by the Cisco IOS software.	distribute-list <i>access-list-number</i> out [<i>interface-name</i> <i>routing-process</i>]
Control the routers from which routing updates are accepted.	ipx router-filter <i>access-list-number</i>

Note The **ipx output-network-filter** command applies to the IPX RIP only. To control the advertising of routes when filtering routing updates in Enhanced IGRP, use the **distribute-list out** command. See the “Control the Advertising of Routes in Routing Updates” section earlier in this chapter for more information.

Create SAP Filters

A common source of traffic on Novell networks is SAP messages, which are generated by NetWare servers and the Cisco IOS software when they broadcast their available services. To control how SAP messages from network segments or specific servers are routed among IPX networks, perform the following steps:

Step 1 Create a SAP access list.

Step 2 Apply one or more filters to an interface.

To create a SAP access list, perform the following task in global configuration mode:

Task	Command
Create a SAP access list.	access-list <i>access-list-number</i> { deny permit } <i>network[.node] [network.node-mask] [service-type</i> <i>[server-name]]</i>

To apply SAP filters to an interface, perform one or more of the following tasks in interface configuration mode:

Task	Command
Filter incoming service advertisements.	ipx input-sap-filter <i>access-list-number</i>
Filter outgoing service advertisements.	ipx output-sap-filter <i>access-list-number</i>
Filter service advertisements received from a particular router.	ipx router-sap-filter <i>access-list-number</i>

You can apply one of each SAP filter to each interface.

For examples of creating and applying SAP filters, see the “SAP Input Filter Example” and “SAP Output Filter Example” sections at the end of this chapter.

Create GNS Response Filters

To create filters for controlling which servers are included in the GNS responses sent by the Cisco IOS software, perform the following tasks:

Step 1 Create a SAP access list.

Step 2 Apply a GNS filter to an interface.

To create a SAP access list, perform the following task in global configuration mode:

Task	Command
Create a SAP access list.	access-list <i>access-list-number</i> { deny permit } <i>network[.node] [network.node-mask] [service-type</i> <i>[server-name]]</i>

To apply a GNS filter to an interface, perform the following task in interface configuration mode:

Task	Command
Filter the list of servers in GNS response messages.	ipx output-gns-filter <i>access-list-number</i>

Create IPX NetBIOS Filters

Novell's IPX NetBIOS allows messages to be exchanged between nodes using alphanumeric names and node addresses. Therefore, the Cisco IOS software lets you filter incoming and outgoing NetBIOS FindName packets by the node name or by an arbitrary byte pattern (such as the node address) in the packet.

Note These filters apply to IPX NetBIOS FindName packets only. They have no effect on LLC2 NetBIOS packets.

Keep the following in mind when configuring IPX NetBIOS access control:

- Host (node) names are case sensitive.
- Host and byte access lists can have the same names because the two types of lists are independent of each other.
- When filtering by node name, the names in the access lists are compared with the destination name field for IPX NetBIOS “find name” requests.
- Access filters that filter by byte offset can have a significant impact on the packet transmission rate because each packet must be examined. You should use these access lists only when absolutely necessary.
- If a node name is not found in an access list, the default action is to deny access.

To create filters for controlling IPX NetBIOS access, perform the following tasks:

Step 1 Create a NetBIOS access list.

Step 2 Apply the access list to an interface.

To create one or more NetBIOS access lists, perform one or both of the following tasks in global configuration mode:

Task	Command
Create an access list for filtering IPX NetBIOS packets by node name.	netbios access-list host <i>name</i> {deny permit} <i>string</i>
Create an access list for filtering IPX NetBIOS packets by arbitrary byte pattern.	netbios access-list bytes <i>name</i> {deny permit} <i>offset</i> <i>byte-pattern</i>

To apply a NetBIOS access list to an interface, perform one or more of the following tasks in interface configuration mode:

Task	Command
Filter incoming packets by node name.	ipx netbios input-access-filter host <i>name</i>
Filter incoming packets by byte pattern.	ipx netbios input-access-filter bytes <i>name</i>
Filter outgoing packets by node name.	ipx netbios output-access-filter host <i>name</i>
Filter outgoing packets by byte pattern.	ipx netbios output-access-filter bytes <i>name</i>

You can apply one of each of these four filters to each interface.

For an example of how to create filters for controlling IPX NetBIOS, see the “IPX NetBIOS Filter Examples” section at the end of this chapter.

Create Broadcast Message Filters

Routers normally block all broadcast requests and do not forward them to other network segments. This is done to prevent the degradation of performance inherent in broadcast traffic over the entire network. You can define which broadcast messages get forwarded to other networks by applying a broadcast message filter to an interface.

To create filters for controlling broadcast messages, perform the following tasks:

Step 1 Create an access list.

Step 2 Apply a broadcast message filter to an interface.

To create an access list, perform one of the following tasks in global configuration mode:

Task	Command
Create a standard IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>source-network</i> [. <i>source-node</i> [<i>source-node-mask</i>]] [<i>destination-network</i> [. <i>destination-node</i> [<i>destination-node-mask</i>]]]
Create an extended IPX access list (for generic, routing, and broadcast filters).	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> [<i>source-network</i> [. <i>source-node</i> [<i>source-network-mask</i> , <i>source-node-mask</i>]]] <i>source-socket</i> [<i>destination-network</i> [. <i>destination-node</i> [<i>destination-network-mask</i> , <i>destination-node-mask</i>]] <i>destination-socket</i>]

To apply a broadcast message filter to an interface, perform the following tasks in interface configuration mode:

Task	Command
Step 1 Specify a helper address for forwarding broadcast messages.	ipx helper-address <i>network.node</i>
Step 2 Apply a broadcast message filter to an interface.	ipx helper-list <i>access-list-number</i>

Note A broadcast message filter has no effect unless you have issued an **ipx helper-address** or an **ipx type-20-propagation** command on the interface to enable and control the forwarding of broadcast messages. These commands are discussed later in this chapter.

For examples of creating and applying broadcast message filters, see the “Helper Facilities to Control Broadcast Examples” section at the end of this chapter.

Tune IPX Network Performance

To tune IPX network performance, perform the tasks in one or more of the following sections:

- Control Novell IPX Compliance
- Configure Static Routes
- Adjust RIP Update Timers
- Configure RIP Update Packet Size

- Configure Static SAP Table Entries
- Configure the Queue Length for SAP Requests
- Adjust SAP Update Timers
- Configure SAP Update Packet Size
- Enable Round-Robin Load Sharing
- Enable Per-Host Load Sharing
- Control Responses to GNS Requests
- Use Helper Addresses to Forward Broadcast Messages
- Enable Fast Switching of IPX Directed Broadcast Packets
- Control the Forwarding of Type 20 Packets
- Disable IPX Fast Switching
- Enable Autonomous Switching
- Enable SSE Switching
- Pad Odd-Length Packets
- Repair Corrupted Network Numbers
- Control Route Cache Size
- Control Route Cache Invalidation

Control Novell IPX Compliance

Cisco's implementation of Novell's IPX protocol is certified to provide full IPX router functionality, as defined by Novell's IPX Router Specification, Version 1.10, published November 17, 1992.

To control specific aspects of IPX compliance, you can use a combination of global configuration and interface configuration commands. You can perform one or more of the following tasks in global configuration mode:

Task	Command
Restrict the acceptance of IPX type 20 propagation packets.	ipx type-20-input-checks
Restrict the forwarding of IPX type 20 propagation packets.	ipx type-20-output-checks
Set the interpacket delay of multiple-packet routing updates sent on all interfaces.	ipx default-output-rip-delay <i>delay</i>
Set the interpacket delay of multiple-packet triggered routing updates sent on all interfaces.	ipx default-triggered-rip-delay <i>delay</i>
Set the interpacket delay of multiple-packet SAP updates sent on all interfaces.	ipx default-output-sap-delay <i>delay</i>
Set the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.	ipx default-triggered-sap-delay <i>delay</i>

You can perform one or more of the following tasks in interface configuration mode:

Task	Command
Set the tick count, which is used in the IPX RIP delay field.	ipx delay <i>number</i>
Administratively shut down an IPX network on an interface. This removes the network from the interface.	ipx down <i>network</i>
Set the interpacket delay of multiple-packet routing updates sent on a single interface.	ipx output-rip-delay <i>delay</i>
Set the interpacket delay of multiple-packet triggered routing updates sent on a single interface.	ipx triggered-rip-delay <i>delay</i>
Set the interpacket delay of multiple-packet SAP updates sent on a single interface.	ipx output-sap-delay <i>delay</i>
Set the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.	ipx triggered-sap-delay <i>delay</i>
Forward IPX type 20 propagation packets to other network segments.	ipx type-20-propagation

Note We recommend that you use an **ipx output-rip-delay** and **ipx output-sap-delay** on slower speed WAN interfaces.

To achieve full compliance, issue the following interface configuration commands on each interface configured for IPX:

Task	Command
Step 1 Set the interpacket delay of multiple-packet routing updates to 55 ms.	ipx output-rip-delay 55
Step 2 Set the interpacket delay of multiple-packet SAP updates to 55 ms.	ipx output-sap-delay 55
Step 3 Optionally enable type 20 packet propagation if you want to forward type 20 broadcast traffic across the router.	ipx type-20-propagation

You can also globally set interpacket delays for multiple-packet RIP and SAP updates to achieve full compliance, eliminating the need to set delays on each interface. To do so, issue the following commands from global configuration mode:

Task	Command
Step 1 Set the interpacket delay of multiple-packet routing updates sent on all interfaces to 55 ms.	ipx default-output-rip-delay 55

Task	Command
Step 2 Set the interpacket delay of multiple-packet SAP updates sent on all interfaces to 55 ms.	ipx default-output-sap-delay 55

Configure Static Routes

IPX uses RIP, Enhanced IGRP, or NLSP to determine the best path when several paths to a destination exist. The routing protocol then dynamically updates the routing table. However, you might want to add static routes to the routing table to explicitly specify paths to certain destinations. Static routes always override any dynamically learned paths.

Be careful when assigning static routes. When links associated with static routes are lost, traffic may stop being forwarded or traffic may be forwarded to a nonexistent destination, even though an alternative path might be available.

To add a static route to the routing table, perform the following task in global configuration mode:

Task	Command
Add a static route to the routing table.	ipx route { <i>network</i> default } { <i>network.node</i> <i>interface</i> } [floating-static]

You can configure static routes that can be overridden by dynamically learned routes. These routes are referred to as floating static routes. You can use a floating static route to create a path of last resort that is used only when no dynamic routing information is available.

Note By default, floating static routes are not redistributed into other dynamic protocols.

To add a floating static route to the routing table, perform the following task in global configuration mode:

Task	Command
Add a floating static route to the routing table.	ipx route { <i>network</i> default } { <i>network.node</i> <i>interface</i> } [floating-static]

Adjust RIP Update Timers

You can set the interval between IPX RIP updates on a per-interface basis. You can also specify the delay between the packets of a multiple-packet RIP update on a per-interface or global basis. Additionally, you can specify the delay between packets of a multiple-packet triggered RIP update on a per-interface or global basis.

You can set RIP update times only in a configuration in which all routers are our routers, or in which the IPX routers allow configurable timers. The timers for all routers connected to the same network segment should be the same. The RIP update value you choose affects internal IPX timers as follows:

- IPX routes are marked invalid if no routing updates are heard within three times the value of the update interval ($3 * interval$) and are advertised with a metric of infinity.
- IPX routes are removed from the routing table if no routing updates are heard within four times the value of the update interval ($4 * interval$).

- If you define an update timer for more than one interface in a router, the granularity of the update timer is determined by the lowest value defined for one of the interfaces in the router. The router “wakes up” at this granularity interval and sends out updates as appropriate. For more information about granularity, see the “Novell IPX Commands” chapter in the *Network Protocols Command Reference, Part 2*.

You might want to set a delay between the packets in a multiple-packet update if there are some slower PCs on the network or on slower-speed interfaces.

To adjust RIP update times on a per-interface basis, perform any or all of the following tasks in interface configuration mode:

Task	Command
Adjust the RIP update interval.	ipx update-time <i>interval</i>
Adjust the delay between multiple-packet routing updates sent on a single interface.	ipx output-rip-delay <i>delay</i>
Adjust the delay between multiple-packet triggered routing updates sent on a single interface.	ipx triggered-rip-delay <i>delay</i>

To adjust RIP update times on a global basis, perform any or all of the following tasks in global configuration mode:

Task	Command
Adjust the delay between multiple-packet routing updates sent on all interfaces.	ipx default-output-rip-delay <i>delay</i>
Adjust the delay between multiple-packet triggered routing updates sent on all interfaces.	ipx default-triggered-rip-delay <i>delay</i>

By default, the RIP entry for a network or server ages out at an interval equal to three times the RIP update interval. To configure the multiplier that controls the interval, perform the following task in interface configuration mode:

Task	Command
Configure the interval at which a network RIP entry ages out.	ipx rip-multiplier <i>multiplier</i>

Configure RIP Update Packet Size

By default, the maximum size of RIP updates sent out an interface is 432 bytes. This size allows for 50 routes at 8 bytes each, plus a 32-byte IPX RIP header. To modify the maximum packet size, perform the following task in interface configuration mode:

Task	Command
Configure the maximum packet size of RIP updates sent out an interface.	ipx rip-max-packetsize <i>bytes</i>

Configure Static SAP Table Entries

Servers use SAP to advertise their services via broadcast packets. The Cisco IOS software stores this information in the SAP table, also known as the Server Information Table. This table is updated dynamically. You might want to explicitly add an entry to the Server Information Table so that clients always use the services of a particular server. Static SAP assignments always override any identical entries in the SAP table that are learned dynamically, regardless of hop count. If a dynamic route that is associated with a static SAP entry is lost or deleted, the software will not announce the static SAP entry until it relearns the route.

To add a static entry to the SAP table, perform the following task in global configuration mode:

Task	Command
Specify a static SAP table entry.	ipx sap <i>service-type name network.node socket hop-count</i>

Configure the Queue Length for SAP Requests

The Cisco IOS software maintains a list of SAP requests to process, including all pending GNS queries from clients attempting to reach servers. When the network is restarted following a power failure or other unexpected event, the router can be inundated with hundreds of requests for servers. Typically, many of these are repeated requests from the same clients. You can configure the maximum length allowed for the pending SAP requests queue. SAP requests received when the queue is full are dropped, and the client must resend them.

To set the queue length for SAP requests, perform the following task in global configuration mode:

Task	Command
Configure the maximum SAP queue length.	ipx sap-queue-maximum <i>number</i>

Adjust SAP Update Timers

You can adjust the interval at which SAP updates are sent. You can also set the delay between packets of a multiple-packet SAP update on a per-interface or global basis. Additionally, you can specify the delay between packets of a multiple-packet triggered SAP update on a per-interface or global basis.

Changing the interval at which SAP updates are sent is most useful on limited-bandwidth, point-to-point links, or on X.25 and Frame Relay multipoint interfaces. You should ensure that all Novell servers and routers on a given network have the same SAP interval. Otherwise, they might decide that a server is down when it is really up.

Adjusting the delay between packets sent in a multiple-packet SAP update is useful when the IPX network has slow IPX servers or routers. Setting a delay between packets in a multiple-packet SAP update forces Cisco routers to slow their output of SAP packets.

To modify the SAP timers on a per-interface basis, perform any or all of the following tasks in interface configuration mode:

Task	Command
Adjust the interval at which SAP updates are sent.	ipx sap-interval <i>interval</i>
Adjust the interpacket delay of multiple-packet SAP updates sent on a single interface.	ipx output-sap-delay <i>delay</i>

Task	Command
Adjust the interpacket delay of multiple-packet triggered SAP updates sent on a single interface.	ipx triggered-sap-delay <i>delay</i>

To adjust SAP update times on a global basis (eliminating the need to configure delays on a per-interface basis), perform any or all of the following tasks in global configuration mode:

Task	Command
Adjust the interpacket delay of multiple-packet SAP updates sent on all interfaces.	ipx default-output-sap-delay <i>delay</i>
Adjust the interpacket delay of multiple-packet triggered SAP updates sent on all interfaces.	ipx default-triggered-sap-delay <i>delay</i>

By default, the SAP entry of a network or server ages out at an interval equal to three times the SAP update interval. To configure the multiplier that controls the interval, perform the following task in interface configuration mode:

Task	Command
Configure the interval at which a network's or server's SAP entry ages out.	ipx sap-multiplier <i>multiplier</i>

Configure SAP Update Packet Size

By default, the maximum size of SAP updates sent out an interface is 480 bytes. This size allows for 7 servers (64 bytes each), plus a 32-byte IPX SAP header. To modify the maximum packet size, perform the following task in interface configuration mode:

Task	Command
Configure the maximum packet size of SAP updates sent out an interface.	ipx sap-max-packetsize <i>bytes</i>

Enable Round-Robin Load Sharing

You can set the maximum number of equal-cost, parallel paths to a destination. (Note that when paths have differing costs, the Cisco IOS software chooses lower-cost routes in preference to higher-cost routes.) The software then distributes output on a packet-by-packet basis in round-robin fashion. That is, the first packet is sent along the first path, the second packet along the second path, and so on. When the final path is reached, the next packet is sent to the first path, the next to the second path, and so on. This round-robin scheme is used regardless of whether fast switching is enabled.

Limiting the number of equal-cost paths can save memory on routers with limited memory or very large configurations. Additionally, in networks with a large number of multiple paths and systems with limited ability to cache out-of-sequence packets, performance might suffer when traffic is split between many paths.

To set the maximum number of paths, perform the following task in global configuration mode:

Task	Command
Set the maximum number of equal-cost paths to a destination.	ipx maximum-paths <i>paths</i>

Enable Per-Host Load Sharing

Round-robin load sharing is the default behavior when you configure **ipx maximum-paths** to a value greater than 1. Round-robin load sharing works by sending data packets over successive equal cost paths without regard to individual end hosts or user sessions. Path utilization is good, but, because packets destined for a given end host may take different paths, they might arrive out of order.

You can address the possibility of packets arriving out of order by enabling per-host load sharing. With per-host load sharing, the router still uses multiple, equal-cost paths to achieve load sharing; however, packets for a given end host are guaranteed to take the same path, even if multiple, equal-cost paths are available. Traffic for different end hosts tend to take different paths, but true load balancing is not guaranteed. The exact degree of load balancing achieved depends on the exact nature of the workload.

To enable per-host load sharing, perform the following tasks in global configuration mode:

Task	Command
Step 1 Set the maximum number of equal cost paths to a destination to a value greater than 1.	ipx maximum-paths <i>paths</i>
Step 2 Enable per-host load sharing.	ipx per-host-load-share

Control Responses to GNS Requests

You can set the method in which the router responds to SAP GNS requests, you can set the delay time in responding to these requests, or you can disable the sending of responses to these requests altogether.

By default, the router responds to GNS requests if appropriate. For example, if a local server with a better metric exists, then the router does not respond to the GNS request on that segment.

The default method of responding to GNS requests is to respond with the server whose availability was learned most recently.

To control responses to GNS requests, perform one or both of the following tasks in global configuration mode:

Task	Command
Respond to GNS requests using a round-robin selection method.	ipx gns-round-robin
Set the delay when responding to GNS requests.	ipx gns-response-delay [<i>milliseconds</i>]

Note The **ipx gns-response-delay** command is also supported as an interface configuration command. To override the global delay value for a specific interface, use the **ipx gns-response-delay** command in interface configuration mode.

You can also disable GNS queries on a per-interface basis. To do so, perform the following task from interface configuration mode:

Task	Command
Disable the sending of replies to GNS queries.	ipx gns-reply-disable

Use Helper Addresses to Forward Broadcast Messages

Routers normally block all broadcast requests and do not forward them to other network segments. This is done to prevent the degradation of performance over the entire network. You can enable the forwarding of broadcast messages (except type 20 broadcasts) to other networks and forward all other unrecognized broadcast messages. These are non-RIP and non-SAP packets that are not addressed to the local network. Forwarding broadcast messages is sometimes useful when a network segment does not have an end-host capable of servicing a particular type of broadcast request. You can specify the address of a server, network, or networks that can process the broadcast messages.

The Cisco IOS software supports all-networks flooded broadcasts (sometimes referred to as *all-nets flooding*). These are broadcast messages that are forwarded to all networks. Use all-nets flooding carefully and only when necessary, because the receiving networks may be overwhelmed to the point that no other traffic can traverse them.

Use the **ipx helper-list** command, described earlier in this chapter, to define access lists that control which broadcast packets get forwarded.

To specify a helper address for forwarding broadcast messages, perform the following task in interface configuration mode:

Task	Command
Specify a helper address for forwarding broadcast messages.	ipx helper-address <i>network.node</i>

You can specify multiple helper addresses on an interface.

For an example of using helper addresses to forward broadcast messages, see the “Helper Facilities to Control Broadcast Examples” section at the end of this chapter.

Enable Fast Switching of IPX Directed Broadcast Packets

By default, Cisco IOS software switches packets that have been helpered to the broadcast address. To enable fast switching of these IPX-directed broadcast packets, perform the following task in global configuration mode:

Task	Command
Enable fast switching of IPX directed broadcast packets.	ipx broadcast-fastswitching

Control the Forwarding of Type 20 Packets

NetBIOS over IPX uses type 20 propagation broadcast packets flooded to all networks to get information about the named nodes on the network. NetBIOS uses a broadcast mechanism to get this information, because it does not implement a network layer.

Routers normally block all broadcast requests. By enabling type 20 packet propagation, IPX interfaces on the router may accept and forward type 20 propagation packets. Before forwarding (flooding) the packets, the router performs loop detection as described by the IPX router specification.

You can configure the Cisco IOS software to apply extra checks to type 20 propagation packets above and beyond the loop detection described in the IPX specification. These checks are the same ones that are applied to helpered all-nets broadcast packets. They can limit unnecessary duplication of type 20 broadcast packets. The extra helper checks are as follows:

- Accept type 20 propagation packets only on the primary network, which is the network that is the primary path back to the source network.
- Forward type 20 propagation packets only via networks that do not lead back to the source network.

While this extra checking increases the robustness of type 20 propagation packet handling by decreasing the amount of unnecessary packet replication, it has two side effects:

- If type 20 packet propagation is not configured on all interfaces, these packets might be blocked when the primary interface changes.
- It might be impossible to configure an arbitrary, manual spanning tree for type 20 packet propagation.

You can enable the forwarding of type 20 packets on individual interfaces, and you can restrict the acceptance and forwarding of type 20 packets. The tasks to do this are described in the following sections.

Enable the Forwarding of Type 20 Packets

By default, type 20 propagation packets are dropped by the Cisco IOS software. You can configure the software to receive type 20 propagation broadcast packets and forward (flood) them to other network segments, subject to loop detection.

To enable the receipt and forwarding of type 20 packets, perform the following task in interface configuration mode:

Task	Command
Forward IPX type 20 propagation packet broadcasts to other network segments.	ipx type-20-propagation

Restrict the Acceptance of Incoming Type 20 Packets

For incoming type 20 propagation packets, the Cisco IOS software is configured by default to accept packets on all interfaces enabled to receive type 20 propagation packets. You can configure the software to accept packets only from the single network that is the primary route back to the source network. This means that similar packets from the same source that are received via other networks will be dropped.

Checking of incoming type 20 propagation broadcast packets is done only if the interface is configured to receive and forward type 20 packets.

To impose restrictions on the receipt of incoming type 20 propagation packets in addition to the checks defined in the IPX specification, perform the following task in global configuration mode:

Task	Command
Restrict the acceptance of IPX type 20 propagation packets.	ipx type-20-input-checks

Restrict the Forwarding of Outgoing Type 20 Packets

For outgoing type 20 propagation packets, the Cisco IOS software is configured by default to send packets on all interfaces enabled to send type 20 propagation packets, subject to loop detection. You can configure the software to send these packets only to networks that are not routes back to the source network. (The software uses the current routing table to determine routes.)

Checking of outgoing type 20 propagation broadcast packets is done only if the interface is configured to receive and forward type 20 packets.

To impose restrictions on the transmission of type 20 propagation packets, and to forward these packets to all networks using only the checks defined in the IPX specification, perform the following task in global configuration mode:

Task	Command
Restrict the forwarding of IPX type 20 propagation packets.	ipx type-20-output-checks

Forward Type 20 Packets Using Helper Addresses

You can also forward type 20 packets to specific network segments using helper addresses rather than using the type 20 packet propagation.

Note Forwarding type 20 packets using helper addresses does not comply with the Novell IPX router specification.

To forward type 20 packets addresses using helper addresses, perform the following task beginning in global configuration mode:

Task	Command
Step 1 Forward IPX type 20 packets to specific networks segments.	ipx type-20-helpered
Step 2 From interface configuration mode, specify a helper address for forwarding broadcast messages, including IPX type 20 packets.	ipx helper-address <i>network.node</i>

The Cisco IOS software forwards type 20 packets to only those nodes specified by the **ipx helper-address** command.

Note Using the **ipx type-20-helpered** command disables the receipt and forwarding of type 20 propagation packets as directed by the **ipx type-20-propagation** command.

Disable IPX Fast Switching

Fast switching allows higher throughput by switching a packet using a cache created by previous packets. Fast switching is enabled by default on all interfaces that support fast switching.

Packet transfer performance is generally better when fast switching is enabled. However, you might want to disable fast switching in order to save memory space on interface cards and to help avoid congestion when high-bandwidth interfaces are writing large amounts of information to low-bandwidth interfaces.



Caution Turning off fast switching increases system overhead.

To disable IPX fast switching, perform the following task in interface configuration mode:

Task	Command
Disable IPX fast switching.	no ipx route-cache

Enable Autonomous Switching

Autonomous switching provides faster packet switching by allowing the ciscoBus controller to switch packets independently without having to interrupt the system processor. It is available only in Cisco 7000 systems. Autonomous switching is disabled by default on all interfaces.

To enable autonomous switching, perform the following task in interface configuration mode:

Task	Command
Enable autonomous switching.	ipx route-cache cbus

Enable SSE Switching

The silicon switching engine (SSE) is on the Silicon Switch Processor (SSP) board in the Cisco 7000 series. SSE switching contributes to very fast packet processing by allowing the SSE to perform switching independently of the system processor.

To enable SSE switching, perform the following task in interface configuration mode:

Task	Command
Enable the SSE switching cache.	ipx route-cache sse

Pad Odd-Length Packets

Some IPX end hosts reject Ethernet packets that are not padded to be an even length. Certain topologies can result in such packets being forwarded onto a remote Ethernet network. Under specific conditions, you can use padding on intermediate media as a temporary workaround for this problem.

To enable the padding of odd-length packets, perform the following tasks in interface configuration mode:

Task	Command
Step 1 Disable fast switching.	no ipx route-cache
Step 2 Enable the padding of odd-length packets.	ipx pad-process-switched-packets

Repair Corrupted Network Numbers

To repair corrupted network numbers on an interface, perform the following tasks in interface configuration mode:

Task	Command
Step 1 Disable fast switching.	no ipx route-cache
Step 2 Repair corrupted network numbers.	ipx source-network-update



Caution The **ipx source-network-update** interface configuration command interferes with the proper working of OS/2 Requestors. Do not use this command in a network that has OS/2 Requestors.



Caution Do not use the **ipx source-network-update** interface configuration command on interfaces on which NetWare servers are using internal network numbers (that is, all 3.1x and 4.0 servers).

Control Route Cache Size

You can limit the number of entries stored in the IPX route cache to free up router memory and aid router processing.

Storing too many entries in the route cache can use a significant amount of router memory, causing router processing to slow. This situation is most common on large networks that run network management applications for NetWare.

For example, if a network management station is responsible for managing all clients and servers in a very large (greater than 50,000 nodes) Novell network, the routers on the local segment can become inundated with route cache entries. You can set a maximum number of route cache entries on these routers to free up router memory and aid router processing.

To set a maximum limit on the number of entries in the IPX route cache, complete this task in global configuration mode:

Task	Command
Set a maximum limit on the number of entries in the IPX route cache.	ipx route-cache max-size <i>size</i>

If the route cache has more entries than the specified limit, the extra entries are not deleted. However, they may be removed if route cache invalidation is in use. See the “Control Route Cache Invalidation” for more information on invalidating route cache entries.

Control Route Cache Invalidation

You can configure the router to invalidate fast switch cache entries that are inactive. If these entries remain invalidated for one minute, the router purges the entries from the route cache.

Purging invalidated entries reduces the size of the route cache, reduces memory consumption, and improves router performance. Also, purging entries helps ensure accurate route cache information.

You specify the period of time that valid fast switch cache entries must be inactive before the router invalidates them. You can also specify the number of cache entries that the router can invalidate per minute.

To configure the router to invalidate fast switch cache entries that are inactive, complete this task in global configuration mode:

Task	Command
Invalidate fast switch cache entries that are inactive.	ipx route-cache inactivity-timeout <i>period [rate]</i>

When you use the **ipx route-cache inactivity-timeout** command with the **ipx route-cache max-size** command, you can ensure a small route cache with fresh entries.

Configure IPX Accounting

IPX accounting allows you to collect information about IPX packets and the number of bytes that are switched through the Cisco IOS software. You collect information based on the source and destination IPX address. Accounting tracks only IPX traffic that is routed through the router; it does not track traffic generated by or terminating at the router itself.

IPX access lists and fast switching support IPX accounting statistics. Autonomous and SSE switching do not support IPX accounting statistics.

The Cisco IOS software maintains two accounting databases: an active database and a checkpointed database.

To enable IPX accounting, perform the following task in interface configuration mode:

Task	Command
Enable IPX accounting.	ipx accounting

To control IPX accounting, perform one or more of the following tasks in global configuration mode:

Task	Command
Set the maximum number of accounting entries.	ipx accounting-threshold <i>threshold</i>
Set the maximum number of transit entries.	ipx accounting-transits <i>count</i>
Filter the networks for which IPX accounting information is kept.	ipx accounting-list <i>number mask</i>

Shut Down an IPX Network

You can administratively shut down an IPX network in two ways. In the first way, the network still exists in the configuration, but is not active. When shutting down, the network sends out update packets informing its neighbors that it is shutting down. This allows the neighboring systems to update their routing, SAP, and other tables without having to wait for routes and services learned via this network to time out.

To shut down an IPX network such that the network still exists in the configuration, perform the following task in interface configuration mode:

Task	Command
Shut down an IPX network, but have the network still exist in the configuration.	ipx down <i>network</i>

In the second way, you shut down an IPX network and remove it from the configuration. To do this, perform one of the following tasks in interface configuration mode:

Task	Command
Shut down an IPX network and remove it from the configuration.	no ipx network
When multiple networks are configured on an interface, shut down all networks and remove them from the interface.	no ipx network <i>network</i> (where <i>network</i> is 1, the primary interface)
When multiple networks are configured on an interface, shut down one of the secondary networks and remove it from the interface.	no ipx network <i>network</i> (where <i>network</i> is the number of the secondary interface [not 1])

When multiple networks are configured on an interface and you want shut down one of the secondary networks and remove it from the interface, perform the second task in the previous table specifying the network number of one of the secondary networks.

Note In future Cisco IOS software releases, primary and secondary networks will not be supported.

For an example of shutting down an IPX network, see the “IPX Routing Example” section at the end of this chapter.

Configure IPX and SPX over WANs

You can configure IPX over dial-on-demand routing (DDR), Frame Relay, Point-to-Point Protocol (PPP), Switched Multimegabit Data Service (SMDS), and X.25 networks. To do this, you configure address mappings as described in the appropriate chapter.

When you configure IPX over PPP, address maps are not necessary for this protocol. Also, you can enable IPX header compression over point-to-point links to increase available useful bandwidth of the link and reduce response time for interactive uses of the link.

You can use fast-switching IPX serial interfaces configured for Frame Relay and SMDS, and you can use fast-switching SNAP-encapsulated packets over interfaces configured for ATM.

Additionally, you can configure the IPXWAN protocol.

For an example of how to configure IPX over a WAN interface, see the “IPX over a WAN Interface Example” section at the end of this chapter.

Configure IPX over DDR

IPX sends periodic watchdog (keepalive) packets. These are keepalive packets that are sent from servers to clients after a client session has been idle for approximately 5 minutes. On a DDR link, this means that a call would be made every 5 minutes, regardless of whether there were data packets to send. You can prevent these calls from being made by configuring the Cisco IOS software to respond to the server’s watchdog packets on a remote client’s behalf. This is sometimes referred to as *spoofing the server*.

When configuring IPX over DDR, you might want to disable the generation of these packets so that a call is not made every 5 minutes. This is not an issue for the other WAN protocols, because they establish dedicated connections rather than establishing connections only as needed.

To keep the serial interface idle when only watchdog packets are being sent, refer to the tasks described in the “Configuring DDR” chapter. For an example of configuring IPX over DDR, see the “IPX over DDR Example” section at the end of this chapter.

Configure SPX Spoofing over DDR

Sequenced Packet Exchange (SPX) sends periodic keepalive packets between clients and servers. Similar to IPX watchdog packets, these are keepalive packets that are sent between servers and clients after the data has stopped being transferred. On pay-per-packet or byte networks, these packets can incur large customer telephone connection charges for idle time. You can prevent these calls from being made by configuring the Cisco IOS software to respond to the keepalive packets on behalf of a remote system.

When configuring SPX over DDR, you might want to disable the generation of these packets so that a call has the opportunity to go idle. This may not be an issue for the other WAN protocols because they establish dedicated connections rather than establishing connections only as needed.

To keep the serial interface idle when only keepalive packets are being sent, refer to the tasks described in the “Configuring DDR” chapter.

For an example of how to configure SPX spoofing over DDR, see the “IPX over DDR Example” section at the end of this chapter.

Configure IPX Header Compression

You can configure IPX header compression over point-to-point links. With IPX header compression, a point-to-point link can compress IPX headers only, or the combined IPX and NetWare Core Protocol headers. Currently, point-to-point links must first negotiate IPX header compression via IPXCP or IXPWAN. The Cisco IOS software supports IPX header compression as defined by RFC 1553.

For details on configuring IPX header compression, refer to the “Configuring SLIP and PPP” chapter in the *Access Services Configuration Guide*.

Configure the IPXWAN Protocol

The Cisco IOS software supports the IPXWAN protocol, as defined in RFC 1634. IPXWAN allows a router that is running IPX routing to connect via a serial link to another router, possibly from another manufacturer, that is also routing IPX and using IPXWAN.

IPXWAN is a connection start-up protocol. Once a link has been established, IPXWAN incurs little or no overhead.

You can use the IPXWAN protocol over PPP. You can also use it over HDLC; however, the devices at both ends of the serial link must be Cisco routers.

To configure IPXWAN, perform the following tasks in interface configuration mode on a serial interface:

Task	Command
Step 1 Ensure that you have not configured an IPX network number on the interface.	no ipx network
Step 2 Enable PPP.	encapsulation ppp ¹
Step 3 Enable IPXWAN.	ipx ipxwan [<i>local-node</i> { <i>network-number</i> unnumbered } <i>local-server-name</i> <i>retry-interval</i> <i>retry-limit</i>]
Step 4 Optionally, define how to handle IPXWAN when a serial link fails.	ipx ipxwan error [reset resume shutdown]
Step 5 Optionally, enable static routing with IPXWAN.	ipx ipxwan static

1. This command is documented in the “Interface Commands” chapter of the *Configuration Fundamentals Command Reference*.

Configure Next Hop Resolution Protocol (NHRP)

Routers, access servers, and hosts can use Next Hop Resolution Protocol (NHRP) to discover the addresses of other routers and hosts connected to a nonbroadcast, multiaccess (NBMA) network. NHRP provides an ARP-like solution that alleviates some NBMA network problems. With NHRP, systems attached to an NBMA network can dynamically learn the NBMA address of the other systems that are part of that network. These systems can then directly communicate without requiring traffic to use an intermediate hop.

For more information on NHRP and Cisco’s implementation, refer to the “Configuring IP” chapter in the *Network Protocols Configuration Guide, Part 1*.

NHRP Configuration Task List

To configure NHRP, perform the tasks described in the following sections. The first task is required, the remainder are optional.

- Enable NHRP on an Interface
- Configure a Station’s Static IPX-to-NBMA Address Mapping
- Statically Configure a Next Hop Server
- Configure NHRP Authentication
- Control NHRP Initiation
- Control NHRP Packet Rate
- Suppress Forward and Reverse Record Options
- Specify the NHRP Responder Address

- Change the Time Period NBMA Addresses Are Advertised as Valid

For NHRP configuration examples, see the “NHRP Example” section at the end of this chapter.

Enable NHRP on an Interface

To enable NHRP for an interface on a router, perform the following task in interface configuration mode. In general, all NHRP stations within a logical NBMA network must be configured with the same network identifier.

Task	Command
Enable NHRP on an interface.	ipx nhrp network-id <i>number</i>

For an example of enabling NHRP, see the “NHRP Example” section at the end of this chapter.

Configure a Station’s Static IPX-to-NBMA Address Mapping

To participate in NHRP, a station connected to an NBMA network must be configured with the IPX and NBMA addresses of its Next Hop Servers. The format of the NBMA address depends on the medium you are using. For example, ATM uses a network-layer service access point (NSAP) address, Ethernet uses a MAC address, and SMDS uses an E.164 address.

These Next Hop Servers are most likely the stations’s default or peer routers, so their IPX addresses are obtained from the station’s network layer forwarding table.

If the station is attached to several link layer networks (including logical NBMA networks), the station should also be configured to receive routing information from its Next Hop Servers and peer routers so that it can determine which IPX networks are reachable through which link layer networks.

To configure static IPX-to-NBMA address mapping on a station (host or router), perform the following task in interface configuration mode:

Task	Command
Configure static IPX-to-NBMA address mapping.	ipx nhrp map <i>ipx-address nbma-address</i>

Statically Configure a Next Hop Server

A Next Hop Server normally uses the network layer forwarding table to determine where to forward NHRP packets and to find the egress point from an NBMA network. A Next Hop Server may alternately be statically configured with a set of IPX address prefixes that correspond to the IPX addresses of the stations it serves, and their logical NBMA network identifiers.

To statically configure a Next Hop Server, perform the following task in interface configuration mode:

Task	Command
Statically configure a Next Hop Server.	ipx nhrp nhs <i>nhs-address</i> [<i>net-number</i>]

To configure multiple networks that the Next Hop Server serves, repeat the **ipx nhrp nhs** command with the same Next Hop Server address, but different IPX network addresses. To configure additional Next Hop Servers, repeat the **ipx nhrp nhs** command.

Configure NHRP Authentication

Configuring an authentication string ensures that only routers configured with the same string can intercommunicate using NHRP. Therefore, if the authentication scheme is to be used, the same string must be configured in all devices configured for NHRP on a fabric. To specify the authentication string for NHRP on an interface, perform the following task in interface configuration mode:

Task	Command
Specify an authentication string.	ipx nhrp authentication <i>string</i>

Control NHRP Initiation

There are two ways you can control when NHRP is initiated:

- Specify which IPX packets trigger an NHRP request.
- Specify how many data packets have been sent to a particular destination before NHRP is attempted.

Both methods are described in this section.

Triggering NHRP by IPX Packet

You can specify an IPX access list that is used to decide which IPX packets trigger the sending of NHRP requests. By default, all non-NHRP packets can trigger NHRP requests. To limit which IPX packets trigger NHRP requests, you must define an access list and then apply it to the interface.

To define an access list, perform one of the following tasks in global configuration mode:

Task	Command
Define a standard IPX access list.	access-list <i>access-list-number</i> { deny permit } <i>source-network</i> [<i>.source-node</i> [<i>source-node-mask</i>]] [<i>destination-network</i> [<i>.destination-node</i> <i>destination-node-mask</i>]]]
Define an extended IPX access list.	access-list <i>access-list-number</i> { deny permit } <i>protocol</i> [<i>source-network</i>] [[[<i>.source-node</i>] <i>source-node-mask</i>] [<i>.source-node</i> <i>source-network-mask.source-node-mask</i>]] [<i>source-socket</i>] [<i>destination.network</i>] [[[<i>.destination-node</i>] <i>destination-node-mask</i>] [<i>.destination-node destination-network-mask</i> <i>.destination-nodemask</i>]] [<i>destination-socket</i>]

Then apply the IPX access list to the interface by performing the following task in interface configuration mode:

Task	Command
Specify an IPX access list that controls NHRP requests.	ipx nhrp interest <i>access-list-number</i>

Triggering NHRP on a Per-Destination Basis

By default, when the software attempts to transmit a data packet to a destination for which it has determined that NHRP can be used, it transmits an NHRP request for that destination. You can configure the system to wait until a specified number of data packets have been sent to a particular destination before NHRP is attempted. To do so, perform the following task in interface configuration mode:

Task	Command
Specify how many data packets are sent to a destination before NHRP is attempted.	ipx nhrp use <i>usage-count</i>

Control NHRP Packet Rate

By default, the maximum rate at which the software sends NHRP packets is 5 packets per 10 seconds. The software maintains a per-interface quota of NHRP packets (whether generated locally or forwarded) that can be transmitted. To change this maximum rate, perform the following task in interface configuration mode:

Task	Command
Change the NHRP packet rate per interface.	ipx nhrp max-send <i>pkt-count every interval</i>

Suppress Forward and Reverse Record Options

To dynamically detect link-layer filtering in NBMA networks (for example, SMDS address screens), and to provide loop detection and diagnostic capabilities, NHRP incorporates a route record in requests and replies. The route record options contain the network (and link layer) addresses of all intermediate Next Hop Servers between source and destination (in the forward direction) and between destination and source (in the reverse direction).

By default, forward record options and reverse record options are included in NHRP request and reply packets. To suppress the use of these options, perform the following task in interface configuration mode:

Task	Command
Suppress forward and reverse record options.	no ipx nhrp record

Specify the NHRP Responder Address

If an NHRP requester wants to know which Next Hop Server generates an NHRP reply packet, it can request that information by including the responder address option in its NHRP request packet. The Next Hop Server that generates the NHRP reply packet then complies by inserting its own IPX address in the NHRP reply. The Next Hop Server uses the primary IPX address of the specified interface.

To specify which interface the Next Hop Server uses for the NHRP responder IPX address, perform the following task in interface configuration mode:

Task	Command
Specify which interface the Next Hop Server uses to determine the NHRP responder address.	ipx nhrp responder <i>type number</i>

If an NHRP reply packet being forwarded by a Next Hop Server contains that Next Hop Server's own IPX address, the Next Hop Server generates an "NHRP Loop Detected" error indication and discards the reply.

Change the Time Period NBMA Addresses Are Advertised as Valid

You can change the length of time that NBMA addresses are advertised as valid in positive and negative NHRP responses. In this context, advertised means how long the Cisco IOS software tells other routers to keep the addresses it is providing in NHRP responses. The default length of time for each response is 7,200 seconds (2 hours). To change the length of time, perform the following task in interface configuration mode:

Task	Command
Specify the number of seconds that NBMA addresses are advertised as valid in positive or negative NHRP responses.	ipx nhrp holdtime <i>seconds-positive</i> [<i>seconds-negative</i>]

Monitor and Maintain the IPX Network

To monitor and maintain a Novell IPX network, perform one or more of the following tasks at the EXEC prompt:

Task	Command
Delete all entries in the IPX accounting or accounting checkpoint database.	clear ipx accounting [checkpoint]
Delete all entries in the IPX fast-switching cache.	clear ipx cache
Delete all NLSP adjacencies from the adjacency database.	clear ipx nlsip [<i>tag</i>] neighbors

Task	Command
Delete entries in the IPX routing table.	clear ipx route [<i>network</i> *]
Have the Cisco 7000 route processor recompute the IPX SSE fast-switching cache.	clear ipx sse
Reinitialize the route processor on the Cisco 7000.	clear sse
List the entries in the IPX accounting or accounting checkpoint database.	show ipx accounting [checkpoint]
List the entries in the IPX fast-switching cache.	show ipx cache
List the neighbors discovered by Enhanced IGRP.	show ipx eigrp neighbors [servers] [<i>autonomous-system-number</i> <i>interface</i>]
Display information about interfaces configured for Enhanced IGRP.	show ipx eigrp interfaces [<i>interface</i>] [<i>as-number</i>]
Display the contents of the Enhanced IGRP topology table.	show ipx eigrp topology [<i>network-number</i>]
Display the status of the IPX interfaces configured in the router and the parameters configured on each interface.	show ipx interface [<i>type number</i>]
Display the entries in the link-state packet (LSP) database.	show ipx nlsp [<i>tag</i>] database [<i>lspid</i>] [detail]
Display the device's NLSP neighbors and their states.	show ipx nlsp [<i>tag</i>] neighbors [<i>interface</i>] [detail]
Display a history of the SPF calculations for NLSP.	show ipx nlsp [<i>tag</i>] spf-log
List the entries in the IPX routing table.	show ipx route [<i>network</i>] [default] [detailed]
List the servers discovered through SAP advertisements.	show ipx servers [unsorted sorted] [name net type] [regex <i>name</i>] ¹
Display information about the number and type of IPX packets transmitted and received.	show ipx traffic
Display a summary of SSP statistics.	show sse summary

1. This command is documented in the “Regular Expressions” appendix of the *Configuration Fundamentals Command Reference*.

The Cisco IOS software can transmit Cisco pings or standard Novell pings as defined in the NLSP specification. By default, the software generates Cisco pings. To choose the ping type, perform the following task in global configuration mode:

Task	Command
Select the ping type.	ipx ping-default { cisco novell }

To initiate a ping, perform one of the following tasks in EXEC mode:

Task	Command
Diagnose basic IPX network connectivity (user-level command).	ping ipx <i>network.node</i>
Diagnose basic IPX network connectivity (privileged command).	ping [ipx] [<i>network.node</i>]

Monitor and Maintain NHRP

To monitor the NHRP cache or traffic, perform either of the following tasks in EXEC mode:

Task	Command
Display the IPX NHRP cache, optionally limited to dynamic or static cache entries for a specific interface.	show ipx nhrp [dynamic static] [type number]
Display NHRP traffic statistics.	show ipx nhrp traffic

The NHRP cache can contain static entries caused by statically configured addresses and dynamic entries caused by the Cisco IOS software learning addresses from NHRP packets. To clear static entries, use the **no ipx nhrp map** command. To clear the NHRP cache of dynamic entries, perform the following task in EXEC mode:

Task	Command
Clear the IPX NHRP cache of dynamic entries.	clear ipx nhrp

Monitor IPX Enhanced IGRP on an IPX Network

To monitor Enhanced IGRP on an IPX network, perform one or more of the following tasks at the EXEC prompt:

Task	Command
List the neighbors discovered by IPX Enhanced IGRP.	show ipx eigrp neighbors [servers] [autonomous-system-number interface]
Display information about interfaces configured for Enhanced IGRP.	show ipx eigrp interfaces [interface] [as-number]
Display the contents of the IPX Enhanced IGRP topology table.	show ipx eigrp topology [network-number]
Display the contents of the IPX routing table, including Enhanced IGRP entries.	show ipx route [network-number]
Display information about IPX traffic, including Enhanced IGRP traffic.	show ipx traffic

Novell IPX Configuration Examples

This section provides configuration examples for the following IPX configuration situations:

- IPX Routing Example
- IPX Routing on Multiple Networks Example
- IPX Routing Protocols Examples
- Enhanced IGRP and NLSP Route Redistribution Example
- NLSP Route Aggregation for Multiple NLSP Version 1.1 Areas Example
- NLSP Route Aggregation for NLSP Version 1.1 and Version 1.0 Areas Example
- NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example
- IPX Enhanced IGRP Example
- Enhanced IGRP SAP Update Examples

- Advertisement and Processing of SAP Update Examples
- IPX Enhanced IGRP Bandwidth Configuration Example
- IPX Network Access Example
- SAP Input Filter Example
- SAP Output Filter Example
- IPX NetBIOS Filter Examples
- Helper Facilities to Control Broadcast Examples
- IPX over a WAN Interface Example
- IPX over DDR Example
- NHRP Example
- NHRP over ATM Example
- IPX Accounting Example

IPX Routing Example

The following configuration commands enable IPX routing, defaulting the IPX host address to that of the first IEEE-conformance interface (in this example, Ethernet 0). Routing is then enabled on Ethernet 0 and Ethernet 1 for IPX networks 2abc and 1def, respectively.

```
ipx routing
interface ethernet 0
  ipx network 2abc
interface ethernet 1
  ipx network 1def
```

IPX Routing on Multiple Networks Example

The following example uses subinterfaces to create four logical networks on Ethernet interface 0. Each subinterface has a different encapsulation. Any interface configuration parameters that you specify on an individual subinterface are applied to that subinterface only.

```
ipx routing
interface ethernet 0.1
  ipx network 1 encapsulation novell-ether
interface ethernet 0.2
  ipx network 2 encapsulation snap
interface ethernet 0.3
  ipx network 3 encapsulation arpa
interface ethernet 0.4
  ipx network 4 encapsulation sap
```

Note When enabling NLSP and configuring multiple encapsulations on the same physical LAN interface, you must use subinterfaces. You cannot use secondary networks.

You can administratively shut down each of the four subinterfaces separately by using the **shutdown** interface configuration command for each subinterface. For example, the following commands administratively shut down a subinterface:

```
interface ethernet 0.3
 shutdown
```

To bring down network 1, use the following commands:

```
interface ethernet 0.1
 ipx down 1
```

To bring network 1 back up, use the following commands:

```
interface ethernet 0.1
 no ipx down 1
```

To remove all the networks on the interface, use the following interface configuration commands:

```
interface ethernet 0.1
 no ipx network
interface ethernet 0.2
 no ipx network
interface ethernet 0.3
 no ipx network
interface ethernet 0.4
 no ipx network
```

Note The following examples discuss primary and secondary networks. In future Cisco IOS software releases, primary and secondary networks will not be supported.

The following example uses primary and secondary networks to create the same four logical networks as shown earlier in this section. Any interface configuration parameters that you specify on this interface are applied to all the logical networks. For example, if you set the routing update timer to 120 seconds, this value is used on all four networks.

```
ipx routing
interface ethernet 0
 ipx network 1 encapsulation novell-ether
 ipx network 2 encapsulation snap secondary
 ipx network 3 encapsulation arpa secondary
 ipx network 4 encapsulation sap secondary
```

Using this method to configure logical networks, if you administratively shut down Ethernet interface 0 using the **shutdown** interface configuration command, all four logical networks are shut down. You cannot bring down each logical network independently using the **shutdown** command; however, you can do this using the **ipx down** command.

To shut down network 1, use the following command:

```
interface ethernet 0
 ipx down 1
```

To bring the network back up, use the following command:

```
interface ethernet 0
 no ipx down 1
```

To shut down all four networks on the interface and remove all the networks on the interface, use one of the following interface configuration commands:

```
no ipx network

no ipx network 1
```

To remove one of the secondary networks on the interface (in this case, network 2), use the following interface configuration command:

```
no ipx network 2
```

The following example enables IPX routing on a FDDI interfaces 0.2 and 0.3. On FDDI interface 0.2, the encapsulation type is SNAP. On FDDI interface 0.3, the encapsulation type is Novell's FDDI_RAW.

```
ipx routing
interface fddi 0.2
ipx network f02 encapsulation snap
interface fddi 0.3
ipx network f03 encapsulation novell-fddi
```

IPX Routing Protocols Examples

Three routing protocols can run over interfaces configured for IPX: RIP, Enhanced IGRP, and NLSP. This section provides examples of how to enable and disable various combinations of routing protocols.

When you enable IPX routing with the **ipx routing** global configuration command, the RIP routing protocol is automatically enabled. The following example enables RIP on networks 1 and 2:

```
ipx routing
!
interface ethernet 0
  ipx network 1
!
interface ethernet 1
  ipx network 2
```

The following example enables RIP on networks 1 and 2 and Enhanced IGRP on network 1:

```
ipx routing
!
interface ethernet 0
  ipx network 1
!
interface ethernet 1
  ipx network 2
!
ipx router eigrp 100
  network 1
```

The following example enables RIP on network 2 and Enhanced IGRP on network 1:

```
ipx routing
!
interface ethernet 0
 ipx network 1
!
interface ethernet 1
 ipx network 2
!
ipx router eigrp 100
 ipx network 1
!
ipx router rip
 no ipx network 1
```

The following example configures NLSP on two of a router's Ethernet interfaces. Note that RIP is automatically enabled on both of these interfaces. This example assumes that the encapsulation type is Ethernet 802.2.

```
ipx routing
 ipx internal-network 3
!
ipx router nlsp area1
 area-address 0 0
!
interface ethernet 0
 ipx network e0 encapsulation sap
 ipx nlsp area1 enable
!
interface ethernet 1
 ipx network e1 encapsulation sap
 ipx nlsp area1 enable
```

Enhanced IGRP and NLSP Route Redistribution Example

The following example configures a router to redistribute NLSP into Enhanced IGRP autonomous system 100 and Enhanced IGRP autonomous system 100 into NLSP:

```
!
ipx router eigrp 100
 redistribute nlsp
!
ipx router nlsp
 redistribute eigrp 100
!
```

NLSP Route Aggregation for Multiple NLSP Version 1.1 Areas Example

The following example shows the route aggregation configuration for a router connecting multiple NLSP version 1.1 areas. In this example, the two areas are *area1* and *area2*. Because both areas are NLSP version 1.1 areas, redistribution of aggregated routes or explicit routes between the two areas is automatic.

```
ipx routing
 ipx internal-network 2000
!
interface ethernet 1
 ipx network 1001
 ipx nlsp area1 enable
!
interface ethernet 2
```

```

ipx network 2001
ipx nlspace area2 enable
!
ipx router nlspace area1
area-address 1000 fffff000
route-aggregation
!
ipx router nlspace area2
area-address 2000 fffff000
route-aggregation

```

NLSP Route Aggregation for NLSP Version 1.1 and Version 1.0 Areas Example

The following example configures the route aggregation feature with customized route summarization. In this example, *area1* is an NLSP version 1.0 area and *area2* is an NLSP version 1.1 area. Any explicit routes learned in *area1* that fall in the range of *aaaa0000* *ffff0000* are redistributed into *area2* as an aggregated route. Explicit routes from *area1* that do not fall in that range are redistributed into *area2* as an explicit route.

Because *area1* is an NLSP version 1.0 area, it cannot accept aggregated routes learned in *area2*. Thus, when redistribution into *area1* occurs, the router sends explicit routes instead of aggregated routes.

```

ipx routing
ipx internal-network 2000
!
interface ethernet 1
ipx network 1001
ipx nlspace area1 enable
!
interface ethernet 2
ipx network 2001
ipx nlspace area2 enable
!
access-list 1200 deny aaaa0000 ffff0000
access-list 1200 permit -1
!
ipx router nlspace area1
area-address 1000 fffff000
!
ipx router nlspace area2
area-address 2000 fffff000
route-aggregation
redistribute nlspace area1 access-list 1200

```

NLSP Route Aggregation for NLSP Version 1.1, Enhanced IGRP, and RIP Example

In the following example, the router connects two NLSP version 1.1 areas, one Enhanced IGRP area, and one RIP area.

Any routes learned via NLSP *a1* that are represented by *aaaa0000* *ffff0000* are not redistributed into NLSP *a2* as explicit routes. Instead, the router generates an aggregated route. Any routes learned via NLSP *a2* that are represented by *bbbb0000* *ffff0000* are not redistributed as explicit routes into NLSP *a1*. Again, the router generates an aggregated route. Any routes learned via RIP that are represented by *cccc0000* *ffff0000* are not redistributed as explicit routes into NLSP *a1* or NLSP *a2*. Instead, the router sends an aggregated route. Likewise, any routes learned via Enhanced IGRP 129 that are represented by *ddd0000* *ffff0000* are not redistributed into NLSP *a1* or NLSP *a2*. Again, the router sends an aggregated route.

```
ipx routing
ipx internal-network 2000
!
interface ethernet 0
ipx network aaaa0000
ipx nlsip a1 enable
!
interface ethernet 1
ipx network bbbb0000
ipx nlsip a2 enable
!
interface ethernet 2
ipx network cccc0000
!
interface ethernet 3
ipx network dddd0000
!
access-list 1200 deny aaaa0000 ffff0000
access-list 1200 permit -1
!
access-list 1201 deny bbbb0000 ffff0000
access-list 1201 permit -1
!
access-list 1202 deny cccc0000 ffff0000
access-list 1202 permit -1
!
access-list 1203 deny dddd0000 ffff0000
access-list 1203 permit -1
!
ipx router nlsip a1
area-address 10000 fffff000
route-aggregation
redistribute nlsip a2 access-list 1201
redistribute rip access-list 1202
redistribute eigrp 129 access-list 1203
!
ipx router nlsip a2
area-address 2000 fffff000
route-aggregation
redistribute nlsip a1 access-list 1200
redistribute rip access-list 1202
redistribute eigrp 129 access-list 1203
!
ipx router eigrp 129
network dddd0000
redistribute nlsip a1
redistribute nlsip a2
```

IPX Enhanced IGRP Example

The following example configures two interfaces for Enhanced IGRP routing in autonomous system 1:

```
ipx routing
!
interface ethernet 0
 ipx network 10
!
interface serial 0
 ipx network 20
!
ipx router eigrp 1
 network 10
 network 20
```

Enhanced IGRP SAP Update Examples

If an Ethernet interface has neighbors that are all configured for Enhanced IGRP, you might want to reduce the bandwidth used by SAP packets by sending SAP updates incrementally. To do this, you would configure the interface as follows:

```
ipx routing
!
interface ethernet 0
 ipx network 10
 ipx sap-incremental eigrp 1
!
interface serial 0
 ipx network 20
!
ipx router eigrp 1
 network 10
 network 20
```

If you want to send periodic SAP updates on a serial line that is configured for Enhanced IGRP and that has an Enhanced IGRP peer on the other sides, use the following commands:

```
ipx routing
!
interface ethernet 0
 ipx network 10
!
interface serial 0
 ipx network 20
 no ipx sap-incremental eigrp 1
!
ipx router eigrp 1
 network 10
 network 20
```

Advertisement and Processing of SAP Update Examples

The following example causes only services from network 3 to be advertised by an Enhanced IGRP routing process:

```
access-list 1010 permit 3
access-list 1010 deny -1
!
ipx router eigrp 100
network 3
distribute-sap-list 1010 out
```

In the following example, the router redistributes Enhanced IGRP into NLSP *areal*. Only services for networks 2 and 3 are accepted by the NLSP routing process.

```
access-list 1000 permit 2
access-list 1000 permit 3
access-list 1000 deny -1
!
ipx router nlsp areal
redistribute eigrp
distribute-sap-list 1000 in
```

IPX Enhanced IGRP Bandwidth Configuration Example

The following example shows how to configure the bandwidth used by IPX Enhanced IGRP. In this example, Enhanced IGRP process 109 is configured to use a maximum of 25 percent (or 32 kbps) of a 128 kbps circuit:

```
interface serial 0
bandwidth 128
ipx bandwidth-percent eigrp 109 25
```

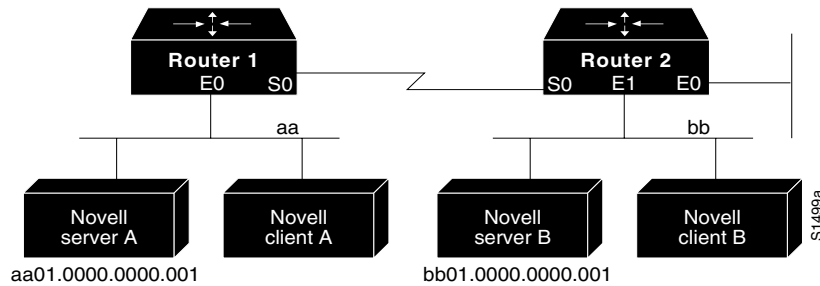
In the following example, the bandwidth of a 56 kbps circuit has been configured to be 20 kbps for routing policy reasons. The Enhanced IGRP process 109 is configured to use a maximum of 200 percent (or 40 kbps) of the circuit.

```
interface serial 1
bandwidth 20
ipx bandwidth-percent eigrp 109 200
```

IPX Network Access Example

Using access lists to manage traffic routing is a powerful tool in overall network control. However, it requires a certain amount of planning and the appropriate application of several related commands. Figure 16 illustrates a network featuring two routers on two network segments.

Figure 16 Novell IPX Servers Requiring Access Control



Suppose you want to prevent clients and servers on Network *aa* from using the services on Network *bb*, but you want to allow the clients and servers on Network *bb* to use the services on Network *aa*. To do this, you would need an access list on Ethernet interface 1 on Router 2 that blocks all packets coming from Network *aa* and destined for Network *bb*. You would not need any access list on Ethernet interface 0 on Router 1.

You would configure Ethernet interface 1 on Router 2 with the following commands:

```
ipx routing
access-list 800 deny aa bb01
access-list 800 permit -1 -1
interface ethernet 1
 ipx network bb
 ipx access-group 800
```

You can accomplish the same result as the previous example more efficiently. For example, you can place the same output filter on Router 1, interface serial 0. Or, you could also place an input filter on interface Ethernet 0 of Router 1, as follows:

```
ipx routing
access-list 800 deny aa bb01
access-list 800 permit -1 -1
interface ethernet 0
 ipx network aa
 ipx access-group 800 in
```

Note When using access control list logging on an interface with fast switching turned on, packets that match the access list (and thus need to be logged) are slow switched, not fast switched.

Logging Access Control List Violations

You can keep a log of all access control list violations by using the keyword **log** at the end of the **access-list** command, as follows:

```
access-list 907 deny -1 -1 0 100 0 log
```

The previous example denies and logs all packets that arrive at the router from any source in any protocol from any socket to any destination on network 100.

The following is an example of a log entry for the **access-list** command:

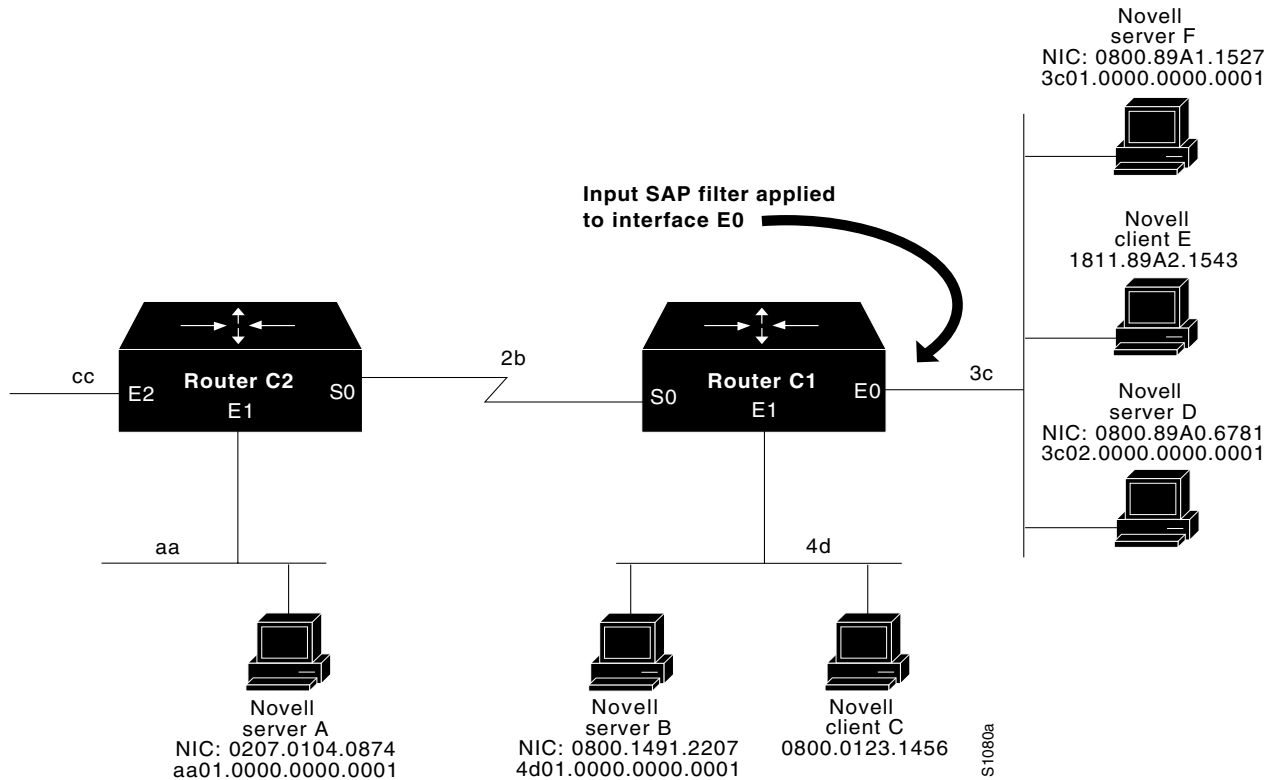
```
%IPX-6-ACL: 907 deny SPX B5A8 50.0000.0000.0001 B5A8 100.0000.0000.0001 10 pkts
```

In this example, 10 SPX packets were denied because they matched access list number 907. The packets were coming from socket B5A8 on networks 50.0000.0000.0001 and were destined for socket B5A8 on network 100.0000.0000.0001.

SAP Input Filter Example

SAP input filters allow a router to determine whether to accept information about a service. Router C1, illustrated in Figure 17, will not accept and, consequently not advertise, any information about Novell server F. However, Router C1 will accept information about all other servers on the network 3c. Router C2 receives information about servers D and B.

Figure 17 SAP Input Filter



The following example configures Router C1. The first line denies server F, and the second line accepts all other servers.

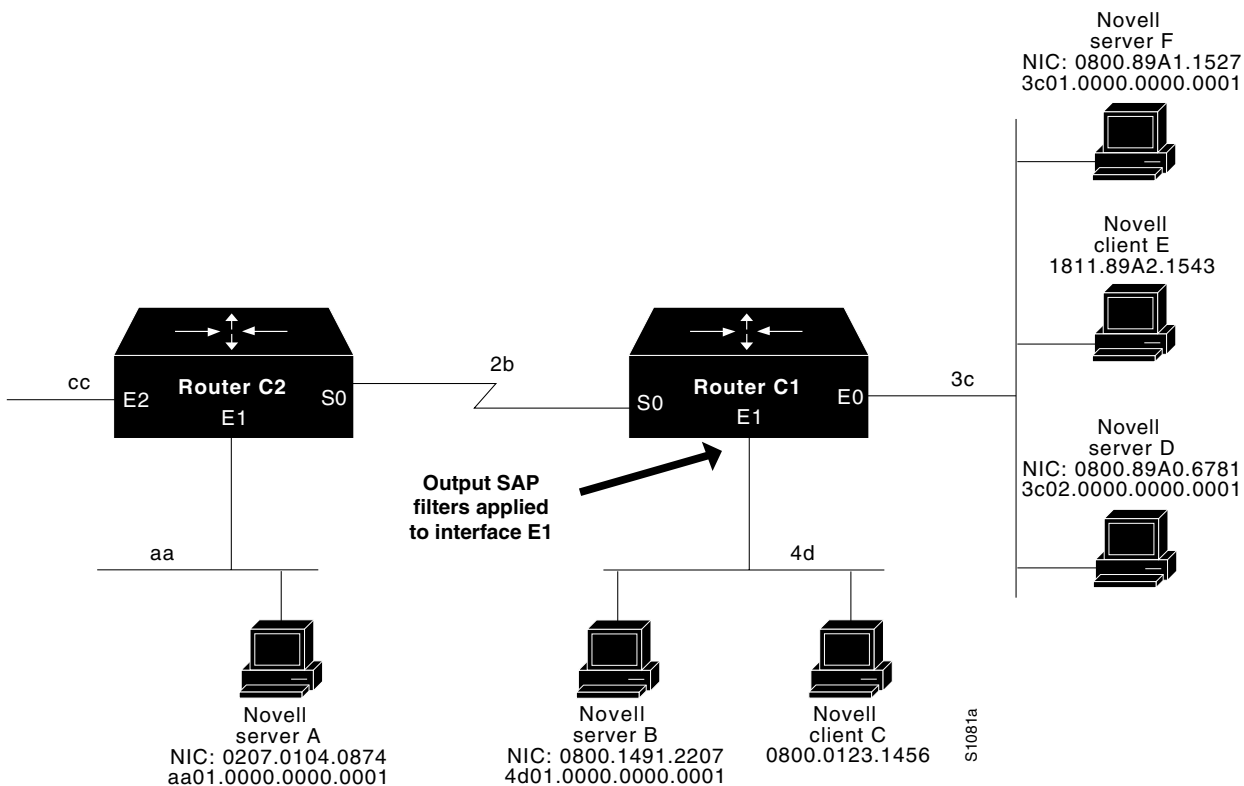
```
access-list 1000 deny 3c01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
 ipx network 3c
 ipx input-sap-filter 1000
interface ethernet 1
 ipx network 4d
interface serial 0
 ipx network 2b
```

Note NetWare Versions 3.11 and later use an internal network and node number as their address for access list commands (the first configuration command in this example).

SAP Output Filter Example

SAP output filters are applied prior to the Cisco IOS software sending information out a specific interface. In the example that follows, Router C1 (illustrated in Figure 18) is prevented from advertising information about Novell server A out interface Ethernet 1, but can advertise server A on network 3c.

Figure 18 SAP Output Filter



The following example refers to Router C1. The first line denies server A. All other servers are permitted.

```
access-list 1000 deny aa01.0000.0000.0001
access-list 1000 permit -1
interface ethernet 0
  novell net 3c
interface ethernet 1
  ipx network 4d
  ipx output-sap-filter 1000
interface serial 0
  ipx network 2b
```

IPX NetBIOS Filter Examples

The following is an example of using a NetBIOS host name to filter IPX NetBIOS frames. The example denies all outgoing IPX NetBIOS frames with a NetBIOS host name of *Boston* on Ethernet interface 0:

```
netbios access-list host token deny Boston
netbios access-list host token permit *
!
ipx routing 0000.0c17.d45d
!
interface ethernet 0
  ipx network 155 encapsulation ARPA
  ipx output-rip-delay 60
  ipx triggered-rip-delay 30
  ipx output-sap-delay 60
  ipx triggered-sap-delay 30
  ipx type-20-propagation
  ipx netbios output-access-filter host token
  no mop enabled
!
interface ethernet 1
  no ip address
  ipx network 105
!
interface fddi 0
  no ip address
  no keepalive
  ipx network 305 encapsulation SAP
!
interface serial 0
  no ip address
  shutdown
!
interface serial 1
  no ip address
  no keepalive
  ipx network 600
  ipx output-rip-delay 100
  ipx triggered-rip-delay 60
  ipx output-sap-delay 100
  ipx triggered-sap-delay 60
  ipx type-20-propagation
```

The following is an example of using a byte pattern to filter IPX NetBIOS frames. This example permits IPX NetBIOS frames from IPX network numbers that end in 05. This means that all IPX NetBIOS frames from Ethernet interface 1 (network 105) and FDDI interface 0 (network 305) will be forwarded by serial interface 0. However, this interface will filter out and not forward all frames from Ethernet interface 0 (network 155).

```

netbios access-list bytes finigan permit 2 **05
!
ipx routing 0000.0c17.d45d
!
ipx default-output-rip-delay 1000
ipx default-triggered-rip-delay 100
ipx default-output-sap-delay 1000
ipx default-triggered-sap-delay 100
!
interface ethernet 0
 ipx network 155 encapsulation ARPA
 ipx output-rip-delay 55
 ipx triggered-rip-delay 55
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
 ipx type-20-propagation
 media-type 10BaseT
!
interface ethernet 1
 no ip address
 ipx network 105
 ipx output-rip-delay 55
 ipx triggered-rip-delay 55
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
 media-type 10BaseT
!
interface fddi 0
 no ip address
 no keepalive
 ipx network 305 encapsulation SAP
 ipx output-sap-delay 55
 ipx triggered-sap-delay 55
!
interface serial 0
 no ip address
 shutdown
!
interface serial 1
 no ip address
 no keepalive
 ipx network 600
 ipx type-20-propagation
 ipx netbios input-access-filter bytes finigan

```

Helper Facilities to Control Broadcast Examples

The following examples illustrate how to control broadcast messages on IPX networks. Note that in the following examples, packet type 2 is used. This type has been chosen arbitrarily; the actual type to use depends on the specific application.

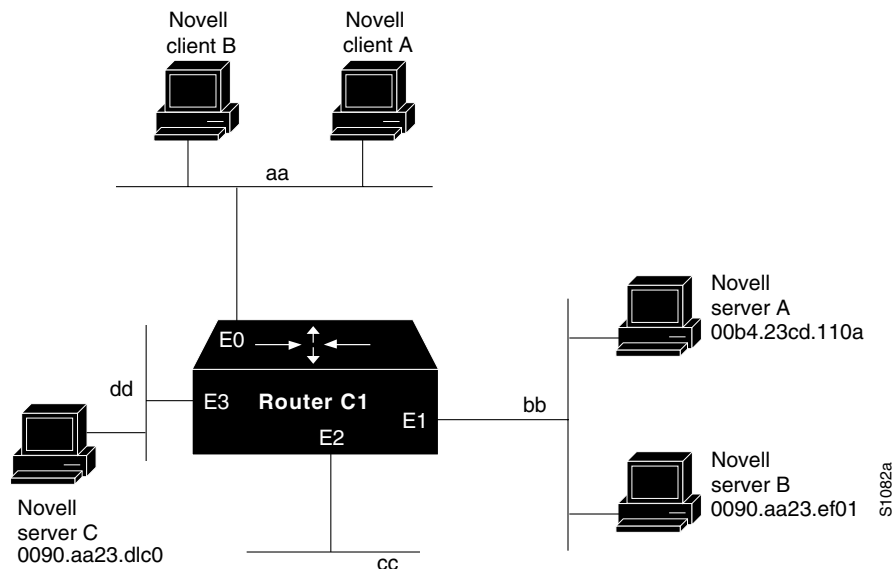
Forwarding to an Address Example

All broadcast packets are normally blocked by the Cisco IOS software. However, type 20 propagation packets may be forwarded, subject to certain loop-prevention checks. Other broadcasts may be directed to a set of networks or a specific host (node) on a segment. The following examples illustrate these options.

Figure 19 shows a router (C1) connected to several Ethernet interfaces. In this environment, all IPX clients are attached to segment *aa*, while all servers are attached to segments *bb* and *dd*. In controlling broadcasts, the following conditions are to be applied:

- Only type 2 and type 20 broadcasts are to be forwarded.
- The IPX clients on network *aa* are allowed to broadcast via type 2 to any server on networks *bb* and *dd*.
- The IPX clients are allowed to broadcast via type 20 to any server on network *dd*.

Figure 19 IPX Clients Requiring Server Access through a Router



The following example configures the router shown in Figure 19. The first line permits broadcast traffic of type 2 from network *aa*. The interface and network commands configure each specific interface. The **ipx helper-address** commands permit broadcast forwarding from network *aa* to *bb* and from network *aa* to *dd*. The helper list allows type 2 broadcasts to be forwarded. (Note that type 2 broadcasts are chosen as an example only. The actual type to use depends on the application.) The **ipx type-20-propagation** command is also required to allow type 20 broadcasts, usually IPX NetBIOS, to be forwarded to all networks where type-20-propagation is enabled. The IPX helper-list filter is applied to both the type 2 packets forwarded by the helper-address mechanism and the type 20 packets forwarded by type-20-propagation.

```
access-list 900 permit 2 aa
interface ethernet 0
  ipx network aa
  ipx type-20-propagation
  ipx helper-address bb.ffff.ffff.ffff
  ipx helper-address dd.ffff.ffff.ffff
  ipx helper-list 900
interface ethernet 1
  ipx network bb
interface ethernet 3
  ipx network dd
  ipx type-20-propagation
```

This configuration means that any network that is downstream from network *aa* (for example, some arbitrary network *aa1*) will not be able to broadcast (type 2) to network *bb* through Router *C1* unless the routers partitioning networks *aa* and *aa1* are configured to forward these broadcasts with a series of configuration entries analogous to the example provided for Figure 19. These entries must be applied to the input interface and be set to forward broadcasts between directly connected networks. In this way, such traffic can be passed along in a directed manner from network to network. A similar situation exists for type 20 packets.

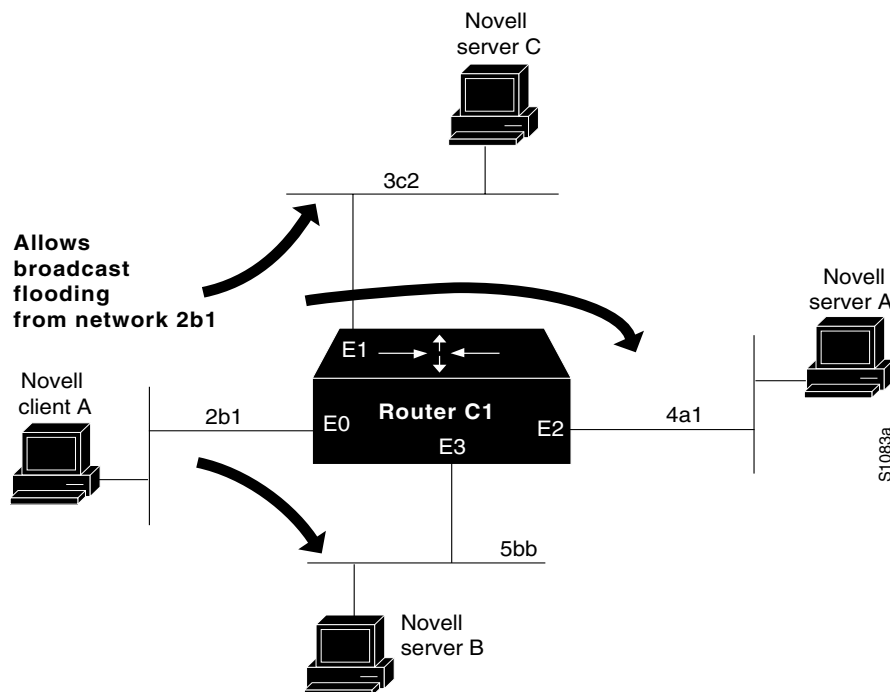
The following example rewrites the **ipx helper-address** interface configuration command line to direct broadcasts to server A:

```
ipx helper-address bb.00b4.23cd.110a
! Permits node-specific broadcast forwarding to
! Server A at address 00b4.23cd.110a on network bb
```

Forwarding to All Networks Example

In some networks, it might be necessary to allow client nodes to broadcast to servers on multiple networks. If you configure your router to forward broadcasts to all attached networks, you are flooding the interfaces. In the environment illustrated in Figure 20, client nodes on network 2b1 must obtain services from IPX servers on networks 3c2, 4a1, and 5bb through Router *C1*. To support this requirement, use the flooding address (-1.ffff.ffff.ffff) in your **ipx helper-address** interface configuration command specifications.

Figure 20 Type 2 Broadcast Flooding



In the following example, the first line permits traffic of type 2 from network 2b1. Then the first interface is configured with a network number. The all-nets helper address is defined and the helper list limits forwarding to type 2 traffic. Type 2 broadcasts from network 2b1 are forwarded to all directly connected networks. All other broadcasts, including type 20, are blocked. To permit broadcasts, delete the **ipx helper-list** entry. To allow type 20 broadcast, enable the **ipx type-20-propagation** interface configuration command on all interfaces.

```
access-list 901 permit 2 2b1
interface ethernet 0
  ipx network 2b1
  ipx helper-address -1.ffff.ffff.ffff
  ipx helper-list 901
interface ethernet 1
  ipx network 3c2
interface ethernet 2
  ipx network 4a1
interface ethernet 3
  ipx network 5bb
```

All-Nets Flooded Broadcast Example

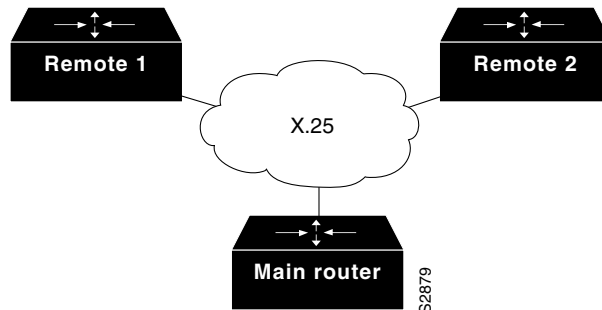
The following example configures all-nets flooding on an interface. As a result of this configuration, Ethernet interface 0 will forward all broadcast messages (except type 20) to all the networks it knows how to reach. This flooding of broadcast messages might overwhelm these networks with so much broadcast traffic that no other traffic may be able to pass on them.

```
interface ethernet 0
  ipx network 23
  ipx helper-address -1.FFFF.FFFF.FFFF
```

IPX over a WAN Interface Example

When you configure the Cisco IOS software to transport IPX packets over a serial interface that is running a WAN protocol such as X.25 or PPP, you specify how the packet will be encapsulated for transport. This encapsulation is not the same as the encapsulation used on an IPX LAN interface. Figure 21 illustrates IPX over a WAN interface.

Figure 21 IPX over a WAN Interface



The following examples configure a serial interface for X.25 encapsulation and for several IPX subinterfaces used in a nonmeshed topology:

Configuration for Main Router

```

hostname Main
!
no ip routing
novell routing 0000.0c17.d726
!
interface ethernet 0
no ip address
Novell network 100
media-type 10BaseT
!
interface serial 0
no ip address
shutdown
!
interface serial 1
no ip address
encapsulation x25
x25 address 33333
x25 htc 28
!
interface serial 1.1 point-to-point
no ip address
novell network 2
x25 map novell 2.0000.0c03.a4ad 11111 BROADCAST
!
interface serial 1.2 point-to-point
no ip address
novell network 3
x25 map novell 3.0000.0c07.5e26 55555 BROADCAST

```

Configuration for Router 1

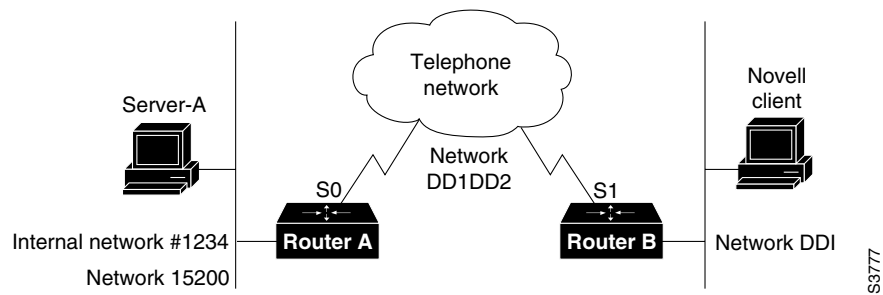
```
hostname Remote1
!
no ip routing
novell routing 0000.0c03.a4ad
!
interface ethernet 0
no ip address
novell network 1
!
interface serial 0
no ip address
encapsulation x25
novell network 2
x25 address 11111
x25 htc 28
x25 map novell 2.0000.0c17.d726 33333 BROADCAST
```

Configuration for Router 2

```
hostname Remote2
!
no ip routing
novell routing 0000.0c07.5e26
!
interface ethernet 0
no ip address
novell network 4
media-type 10BaseT
!
interface serial 0
no ip address
shutdown
!
interface serial 1
no ip address
encapsulation x25
novell network 3
x25 address 55555
x25 htc 28
x25 map novell 3.0000.0c17.d726 33333 BROADCAST
```

IPX over DDR Example

In the configuration shown in Figure 22, an IPX client is separated from its server by a DDR telephone line.

Figure 22 IPX over DDR Configuration

Routing and service information is sent every minute. The output RIP and SAP filters defined in this example filter these updates, preventing them from being sent between Routers A and B. If you were to forward these packets, the two routers would each have to telephone the other once a minute. On a serial link that charges based on the number of packets transmitted, this is generally not desirable. This might not be an issue on a dedicated serial line.

Once the server and client have established contact, the server will send keepalive (watchdog) packets regularly. When SPX is used, both the server and the client send keepalive packets. The purpose of these packets is to ensure that the connection between the server and the client is still functional; these packets contain no other information. Servers send watchdog packets approximately every 5 minutes.

If you were to allow Router A to forward the server's keepalive packets to Router B, Router A would have to telephone Router B every 5 minutes just to send these packets. Again, on a serial link that charges based on the number of packets transmitted, this is generally not desirable. Instead of having Router A telephone Router B only to send keepalive packets, you can enable watchdog spoofing on Router A. This way, when the server connected to this router sends keepalive packets, Router A will respond on behalf of the remote client (the client connected to Router B). When SPX is used, you must enable spoofing of SPX keepalive packets on both Router A and Router B to inhibit the sending of them because both the server and the client send keepalive packets.

Configuration for Router A

```
novell routing 0000.0c04.4878
!
interface Ethernet0
  novell network 15200
!
interface Serial0
!ppp encap for DDR(recommended)
  encapsulation ppp
  novell network DD1DD2
!kill all rip updates
  novell output-network-filter 801
!kill all sap updates
  novell output-sap-filter 1001
! fast-switching off for watchdog spoofing
  no novell route-cache
!don't listen to rip
  novell router-filter 866
!novell watchdog spoofing
  novell watchdog-spoof
!SPX watchdog spoofing
  ipx spx-spoof
!turn on DDR
  dialer in-band
```

```

dialer idle-timeout 200
dialer map IP 198.92.96.132 name R13 7917
dialer map NOVELL DD1DD2.0000.0c03.e3c3 7917
dialer-group 1
ppp authentication chap
!chap authentication required
pulse-time 1
!
access-list 801 deny FFFFFFFF
access-list 866 deny FFFFFFFF
!serialization packets
access-list 900 deny 0 FFFFFFFF 0 FFFFFFFF 457
!RIP packets
access-list 900 deny 1 FFFFFFFF 453 FFFFFFFF 453
!SAP packets
access-list 900 deny 4 FFFFFFFF 452 FFFFFFFF 452
!permit everything else
access-list 900 permit -1 FFFFFFFF 0 FFFFFFFF 0
!
access-list 1001 deny FFFFFFFF
!
!static novell route for remote network
novell route DD1 DD1DD2.0000.0c03.e3c3
!
!
!IPX will trigger the line up (9.21 and later)
dialer-list 1 list 900

```

Configuration for Router B

```

novell routing 0000.0c03.e3c3
!
interface Ethernet1/0
novell network DD1
!
interface Serial2/0
encapsulation ppp
novell network DD1DD2
novell output-network-filter 801
novell output-sap-filter 1001
no novell route-cache
novell router-filter 866
ipx spx-spoof
dialer in-band
dialer idle-timeout 200
dialer map IP 198.92.96.129 name R5 7919
dialer map NOVELL DD1DD2.0000.0c04.4878 7919
dialer-group 1
ppp authentication chap
pulse-time 1
!
access-list 801 deny -1
access-list 866 deny -1
access-list 900 deny 0 FFFFFFFF 0 FFFFFFFF 457
access-list 900 deny 1 FFFFFFFF 453 FFFFFFFF 453
access-list 900 deny 4 FFFFFFFF 452 FFFFFFFF 452
access-list 900 permit -1 FFFFFFFF 0 FFFFFFFF 0
access-list 1001 deny FFFFFFFF
!
!static novell route for server's internal network
novell route 1234 DD1DD2.0000.0c04.4878
novell route 15200 DD1DD2.0000.0c04.4878
!static route
!The following line is the static novell sap required to get to the remote server.

```

```

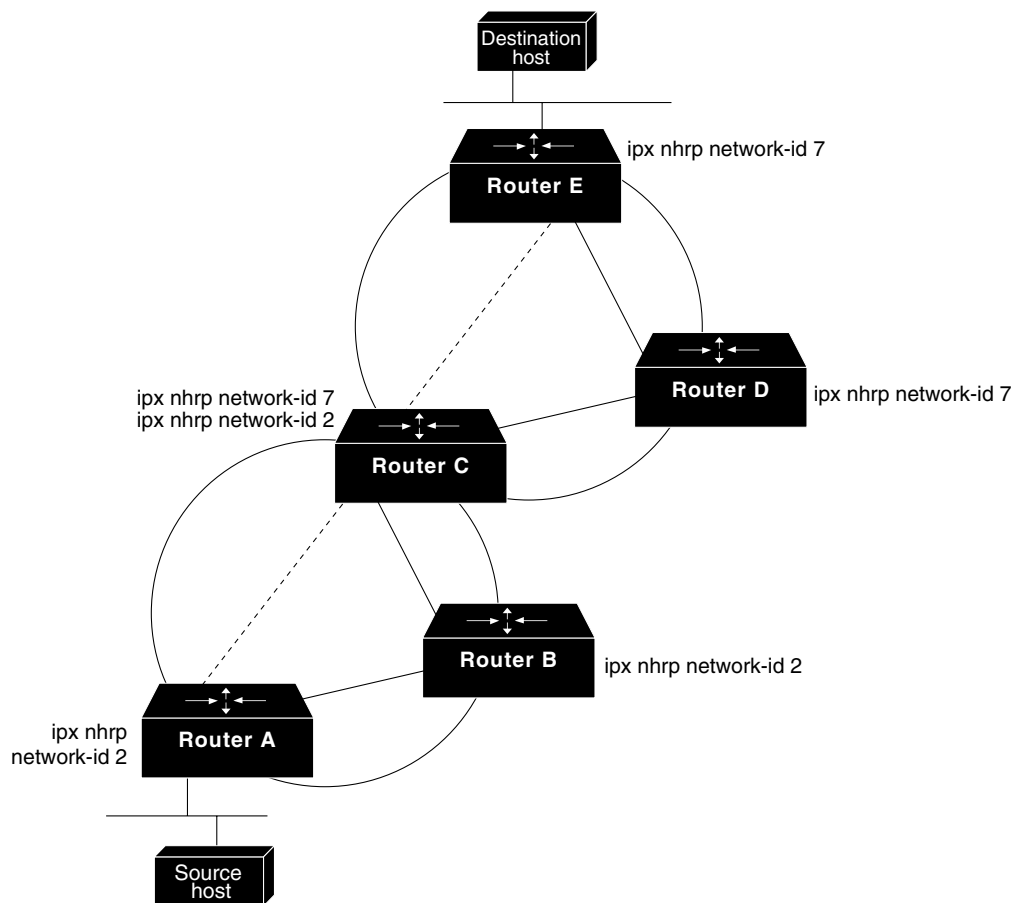
!It informs the router of the next hop.
novell sap 4 CE1-LAB 1234.0000.0000.0001 451 4 <====
!
dialer-list 1 list 900

```

NHRP Example

A logical NBMA network is considered the group of interfaces and hosts participating in NHRP and having the same network identifier. Figure 23 illustrates two logical NBMA networks (shown as circles) configured over a single physical NBMA network. Router A communicates with Routers B and C because they share the same network identifier (2). Router C also communicates with Routers D and E because they share network identifier 7. After address resolution is complete, Router A sends IPX packets to Router C in one hop, and Router C sends them to Router E in one hop, as shown by the dotted lines.

Figure 23 Two Logical NBMA Networks over One Physical NBMA Network



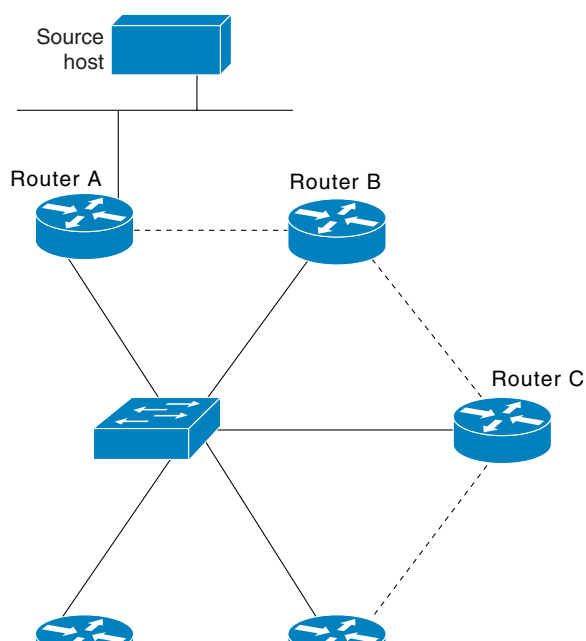
—— = Statically configured tunnel end points or permanent virtual circuits

----- = Dynamically created virtual circuits

63906

The physical configuration of the five routers in Figure 23 might actually be that shown in Figure 24. The source host is connected to Router A and the destination host is connected to Router E. The same switch serves all five routers, making one physical NBMA network.

Figure 24 Physical Configuration of a Sample NBMA Network



Refer again to Figure 23. Initially, before NHRP resolves any NBMA addresses, IPX packets from the source host to the destination host travel through all five routers connected to the switch before reaching the destination. When Router A first forwards the IPX packet toward the destination host, Router A also generates an NHRP request for the destination host's IPX address. The request is forwarded to Router C, where a reply is generated. Router C replies because it is the egress router between the two logical NBMA networks.

Similarly, Router C generates an NHRP request of its own, to which Router E replies. In this example, subsequent IPX traffic between the source and the destination still requires two hops to traverse the NBMA network because the IPX traffic must be forwarded between the two logical NBMA networks. Only one hop would be required if the NBMA network was not logically divided.

NHRP over ATM Example

The following example shows a configuration of three routers using NHRP over ATM. Router A is configured with a static route, which it uses to reach the IPX network where Router B resides. Router A initially reaches Router B through Router C. Router A and Router B directly communicate without Router C once NHRP resolves Router A's and Router C's respective NSAP addresses.

The significant portions of the configurations for Routers A, B, and C follow:

Router A

```
interface ATM0/0
  map-group a
  atm nsap-address 11.1111.11.111111.1111.1111.1111.1111.1111.11
  atm rate-queue 1 10
  atm pvc 1 0 5 qsaal
  ipx network 1
  ipx nhrp network-id 1

map-list a
ipx 1.0000.0c15.3588 atm-nsap 33.3333.33.333333.3333.3333.3333.3333.3333.33

ipx route 2 1.0000.0c15.3588
```

Router B

```
interface ATM0/0
  map-group a
  atm nsap-address 22.2222.22.222222.2222.2222.2222.2222.2222.22
  atm rate-queue 1 10
  atm pvc 2 0 5 qsaal
  ipx network 2
  ipx nhrp network-id 1

map-list a
ipx 2.0000.0c15.3628 atm-nsap 33.3333.33.333333.3333.3333.3333.3333.3333.33

ipx route 1 2.0000.0c15.3628
```

Router C

```

interface ATM0/0
  atm rate-queue 1 10
  atm pvc 2 0 5 qsaa1

interface ATM0/0.1 multipoint
  map-group a
  atm nsap-address 33.3333.33.333333.3333.3333.3333.3333.3333.33
  ipx network 1
  ipx nhrp network-id 1

interface ATM0/0.2 multipoint
  map-group b
  atm nsap-address 33.3333.33.333333.3333.3333.3333.3333.3333.33
  ipx network 2
  ipx nhrp network-id 2

map-list a
  ipx 1.0000.0c15.4f80 atm-nsap 11.1111.11.111111.1111.1111.1111.1111.1111.11

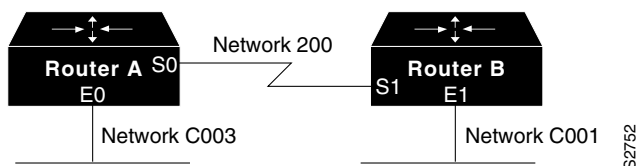
map-list b
  ipx 2.0000.0c15.5021 atm-nsap 22.2222.22.222222.2222.2222.2222.2222.2222.22

```

IPX Accounting Example

The following example configures two Ethernet network segments that are connected via a serial link (see Figure 25). On Router A, IPX accounting is enabled on both the input and output interfaces (that is, on Ethernet interface 0 and serial interface 0). This means that statistics are gathered for traffic traveling in both directions (that is, out to the Ethernet network and out the serial link). However, on Router B, IPX accounting is enabled only on the serial interface and not on the Ethernet interface. This means that statistics are gathered only for traffic that passes out the router on the serial link.

Figure 25 IPX Accounting Example



Configuration for Router A

```

ipx routing
interface ethernet 0
  no ip address
  ipx network C003
  ipx accounting
interface serial 0
  no ip address
  ipx network 200
  ipx accounting

```

Configuration for Router B

```
ipx routing
interface ethernet 1
  no ip address
  no keepalive
  ipx network C001
  no mop enabled
interface serial 1
  no ip address
  ipx network 200
  ipx accounting
```

